



60

***BUSINESS
APPLICATIONS
PROGRAMS***

FOR THE TRS-80 MODEL 100 COMPUTER

***TERRY KEPNER
MARK ROBINSON***

60 BUSINESS APPLICATIONS PROGRAMS

FOR THE TRS-80 MODEL 100 COMPUTER

***TERRY KEPNER
MARK ROBINSON***

Scott, Foresman and Company

Glenview, Illinois

London

To Richard Staron, for starting me along the path to
getting this book written.

Terry Kepner

To Wayne Green, who during idle conversation
convinced me to subscribe to issue 1 of his new
magazine, *Kilobaud Microcomputing*, about these
strange new machines called *computers*.

Mark Robinson

ISBN 0-673-18052-2

Copyright © 1985 Scott, Foresman and Company.

All Rights Reserved.

Printed in the United States of America.

Library of Congress Cataloging in Publication Data

Kepner, Terry.

60 business applications programs for the TRS-80
Model 100 computer.

Includes index.

1. TRS-80 Model 100 (Computer)—Programming.
2. Business—Data processing. I. Robinson, Mark, 1955- . II. Title. III. Title: Sixty business applications programs for the TRS-80 Model 100 computer.

HF5548.4.T75K46 1985 650'.028'5425 84-13890

ISBN 0-673-18052-2

1 2 3 4 5 6-KPF-89 88 87 86 85 84

Radio Shack, TRS-80, and TRS-80 Model 100 Portable Computer are
registered trademarks of the Tandy Corporation, Fort Worth, Texas.

Notice of Liability

The information in this book is distributed on an "AS IS" basis, without warranty. Neither the author nor Scott, Foresman and Company shall have any liability to customer or any other person or entity with respect to any liability, loss, or damage caused or alleged to be caused directly or indirectly by the programs contained herein. This includes, but is not limited to, interruption of service, loss of data, loss of business or anticipatory profits, or consequential damages from the use of the programs.

Preface

The Model 100 computer is a powerful, portable tool for the businessperson on the go. Its built-in text processor, schedule file, address file, and telecommunications program were all designed to give the computer as wide an appeal as possible. Since the computer is aimed at the general market and not just one specific branch, Radio Shack included the BASIC programming language so that people in specific fields could write their own programs, supplementing the built-in programs and giving the owner a customized tool for his or her business.

Unfortunately, most purchasers of the Model 100 computer know nothing about computer programming, and the majority don't want to learn how to program. Instead, they want to concentrate on just using the machine. This is a reasonable approach given the problems of programming proficiently. Learning to program is like learning a new language: it takes time to learn the meaning of the words and the proper methods of stringing them together. If you're trying to do it yourself without an instructor to help, the task becomes Sisyphean. As a general rule, it takes about a year of study and trial-and-error work before the student is familiar enough with the BASIC language to write an efficient and working program in a short amount of time.

This has generated a *program gap*. People are using the computer's built-in programs and searching for knowledgeable programmers to supply the BASIC programs needed for really efficient use of the computer. This doesn't always work, as there are more computers

than good programmers. Hence, many people try to write their own programs, with varying degrees of success.

This program gap led directly to this book. Mark is a business-person-programmer, and I'm a writer-programmer. We decided that the best thing to do to alleviate this lack of business programs was to create a book that supplied them, pooling his knowledge of business, and the attendant formulas, with my writing and programming skills.

Mark wrote each program core, giving the basic input/output code and business formula. I took this core, checked it, cleaned it up, and wrote the text explanation and examples. The result is the best of both worlds. The programs are designed by a businessperson using his knowledge of what a businessperson wants and needs, and the programs are written by a professional writer and programmer so that they are efficient and easy to use and understand.

TERRY KEPNER

Contents

	How to Use This Book*	viii
1	Management Utilities	1
	Ratios	2
	Calculator	20
	Breakeven Unit Analysis	30
	Invoice	36
	Job Quote	49
	Calendar	58
	Days between Dates	62
2	Management Decision Making	65
	Term versus Whole Life Insurance	66
	Time Payment versus Purchase	71
	Total Interest Paid on a Loan	76
	Converting Add-on Interest to Annual Interest	79
	Loan Repayment and Rebate	82
	Purchase versus Lease	86
	Installment Plan Schedule	92
	Capital Asset Pricing Model	96
	Cash Management Model	100
	Price before Sales Tax	104
	Markup (Calculating Retail Price)	107
	Cost before Markup	110

*Includes menu program.

Percentage Markup	113
Discounted Price	116
Price before Discount	119
Profit Margin	122
Pie Chart	125
Economic Order Quantity	130
Bar Chart	133
3 Accountant's Helpers	141
Depreciation	142
Straight-line Depreciation	143
Sum-of-Year's-Digits Depreciation	146
Declining-Balance Depreciation	150
Straight-line versus Declining-Balance Depreciation	155
Amortization	158
Average Yield of an Investment	164
Weighted Average Number of Stock Shares Outstanding	168
4 Time Is Money	175
Simple Interest—Future Value ($PV, I, N = FV$)	177
Compound Interest—Future Value ($PV, I, N = FV$)	181
Other Compound Interest Computations	185
Compound Interest—Time Period ($PV, I, FV = N$)	186
Compound Interest—Present Value ($FV, I, N = PV$)	190
Compound Interest—Interest Rate ($PV, FV, N = I$)	194
Equivalent Interest—Comparing Compound Periods ($IR(c) = IR(c)$)	197
Annuities, Ordinary and Due	201
Ordinary Annuity—Future Value ($PMT, N, I = FV$)	202
Ordinary Annuity—Future Value with Continuous Compounding of Interest, PV Known ($PV, N, I = FV$)	205
Ordinary Annuity—Future Value with Continuous Compounding of Interest, PMT Known ($PMT, N, I = FV$)	208

Ordinary Annuity—Present Value (PMT,N,I = PV)	211
Ordinary Annuity—Present Value with Continuous Compounding of Interest, PMT Known (PMT,N,I = PV)	214
Ordinary Annuity—Present Value with Continuous Compounding of Interest, FV Known (FV,N,I = PV)	217
Ordinary Annuity—Payment, PV Known (PV,N,I = PMT)	220
Ordinary Annuity—Payment, FV Known (FV,N,I = PMT)	224
Ordinary Annuity—Interest Rate, FV Known (PMT,N,FV = I)	227
Ordinary Annuity—Interest Rate, PV Known (PMT,N,PV = I)	230
Ordinary Annuity—Number of Payments (PV,PMT,I = N)	233
Annuity Due—Future Value (I,PMT,N = FV)	237
Annuity Due—Present Value (I,PMT,N = PV)	242
Annuity Due—Payment (PV,N,I = PMT)	245
Annuity Due—Interest Rate, PV Known (PV,PMT,N = I)	249
Annuity Due—Interest Rate, FV Known (FV,PMT,N = I)	252
5 Programming Utilities	255
PACKER	256
PURGER	261
Select Bibliography	265
Glossary	267
Index	273

How to Use This Book

You'll notice that the programs in this book are sequentially numbered. For example, in the Time Is Money chapter (chapter four), the first program's lines are numbered 200 through 240, the second program's line numbers are 250 through 325, the third program's line numbers are 350 through 440, and so forth. (Well, that's not precisely correct: we had to reorganize the programs into a more coherent order after most of the book was already written; thus, the programs aren't in straight-sequence order, but the principle has been kept intact.) This makes it possible for you to mix and match the programs, merging the programs you need most frequently into one large program. This frees up room in the Model 100 directory.

For example, the first ten programs in the Time Is Money chapter occupy only about 10K of RAM; by combining them into one long program you use only one entry space in the directory instead of ten, freeing up the other nine for other programs or data files.

In order to get this merging technique to work, you have to write a short *menu program* that lets you select the particular routine you want when you run the program. A sample menu program is shown below:

```
10 CLS:PRINT" 1. Simple Interest":PRINT" 2.  
Compound Interest":PRINT" 3. Equivalent Interest  
Rates":PRINT" 4. Time Computation":PRINT" 5.
```



```
Present value":PRINT" 6. MENU"  
20 INPUT"Your Choice";A$:A=VAL(A$):  
IFA<10RA>6THEN10  
30 ONAGOTO 200,250,350,450,550  
40 MAXFILES=0:CLEAR0:MENU
```

To make your own menu, just put in the names of the programs, change the 6 in line 20 to match the high number you use, and use the first line number of each program in place of the numbers I used in line 30. The only qualification is that the menu program lines must be numbered between 1 and 199 (that's why the first program starts with line number 200: to leave room for a menu program, if it's needed).

The last choice listed, 6, returns you to the Model 100 menu. The ON . . . GOTO statement has only five line numbers in it, so by selecting option 6 you skip the ON . . . GOTO statement, and line 40 is executed. The MAXFILES command removes any input/output buffers that may have been used, freeing up additional memory for other uses. Similarly, the CLEAR also frees up memory.

If you have already typed in the programs as separate units, you can easily merge them together: load each one into BASIC and save it to RAM as a .DO file, type in the menu program in BASIC, and then type MERGE"filename.DO" for each of the program units you saved previously. BASIC will automatically add each one to the program in BASIC. When you're finished, all the programs will have been made into one big program.

For example, if you want the first five programs mentioned in my sample program merged, you would type in the first program, Simple Interest (A1), and then type SAVE"A1.DO". Next, type in the second program, Compound Interest (A2); then type SAVE"A2.DO". Repeat this for the other three programs: Equivalent Interest Rate (A6), Time Computation (A3), and Present Value (A4). When you look at your Model 100 menu you should see all five programs stored as text files: i.e., A1.DO, A2.DO, A3.DO, A4.DO, and A6.DO.

Now go back to BASIC, and type in your menu program, referring to the text files of the programs for the proper starting line numbers to use in the ON . . . GOTO statement. When you're through, type MERGE"A1.DO".

The computer will respond by flashing the message WAIT at the bottom of your display. Each time the WAIT message changes from normal display to reverse or vice versa, one line of your text

file has been successfully merged with BASIC. Finally, the display will remove the WAIT message and put OK under your merge command, signifying that BASIC is ready for your next instruction. Type MERGE"A2.DO", and the process will repeat. Keep merging the programs until you are finished.

The only error message you could see is *Direct Statement in File*, which means that one of the lines of your text file does not start with a line number. Simply list your program, and see how far BASIC went before it ran into the error. Then go to your text file, and fix the problem. Return to BASIC, and merge the program again.

Sometimes you might not have enough room for all the programs as .BA files and .DO files, and the bigger program you're trying to make. In that case, just convert and merge the programs one at a time, deleting the .DO file after each successful merge.

Another point you will notice is that all the programs have lines numbered 10000 and 10005. These two lines let you rerun the program or return to the Model 100 menu. The reason they are numbered the same in all the programs is so that, when you merge programs, you don't waste space with needless repetitive code.

Line 10005 is similar to line 40 in the menu program in that it returns you to the Model 100 menu, setting MAXFILES to zero and clearing string space first. To use this with a larger, menu-driven program, you'll want to return from the individual program segments to the programs menu, not the Model 100 menu. So, after making—merging—your larger program, edit line 10005, remove the commands MAXFILES=0:CLR0:MENU, and replace them with RUN.

This change will fix it so that line 10005 will return you either to the main program menu or to the program segment you just used. The program main menu will have an option to return you to the Model 100 menu.

As you examine the programs in this book, you'll see other routines that are repetitive, with line numbers between 9000 and 9999. These don't need to be changed when merged with a larger program.

You'll also note that occasionally within a program the word REM or the apostrophe (') appears. The text that follows either one of these is a comment only and doesn't affect the operation of the program. You can leave these comments out of programs without affecting them. The comments were put in to make it easier for you to understand how the programs work.

If you want to experiment with the programs, adding, deleting, or just compressing them, we have provided a line number and variable name cross-reference list for each program.

The cross-reference is easy to use. First are the line numbers referred to by GOTO, GOSUB, THEN, and ELSE statements. Next are listed the variable names used by the program, with the line numbers on which they appear. For example, the cross-reference for the first Time Is Money program, Simple Interest, is reproduced below:

<i>Ref</i>	<i>Line Numbers</i>
:00200:	240
:10000:	240
:I :	215 230
:	Interest rate
:P :	205 230 235
:	Principal
:P\$:	210 215 220 225 235
:	Interest period
:T :	225 230 235
:	Number of interest periods

This says that line 200 is referenced, or *called*, by a command in line 240. Line 10000 is referenced by a command in line 240. The variable name I, storing the interest rate, is used in lines 215 and 230; P, the principal amount, is used in lines 205, 230, and 235. P\$, the interest period, is used in lines 210, 215, 220, 225, and 235. And T, the number of interest periods to calculate, is used in lines 225, 230, and 235.

Knowing which lines are referenced by GOTO, GOSUB, THEN, and ELSE statements, as well as which variable names are in use, where they are used, and what the variables store, makes it easier for someone to modify a program. Continuing with the example of the Simple Interest program: I know, from the cross-reference, that lines 205 through 240 can all be combined into one line without causing a problem. But if I accidentally eliminate line 200 or line 10000, then I will get a *UL error* (undefined line) when the commands in line 240 try to transfer control to the nonexistent line 200 or 10000.

The Simple Interest program can be compressed to look like this:


```
200 CLS:PRINTTAB(13)"SIMPLE INTEREST":REM
    A1
205 PRINT:INPUT"Principal
(dollars)";P:PRINT"Interest rate (%per
";P$;"")";INPUTI;P$=P$+"s":PRINT"Number of
";P$;:INPUTT:P=P+(T*(P*(I/100))):PRINT"Total
after";T;P$;" is
";:PRINTUSING"$$#,#####.##";P:GOSUB1000:RUN200
10000 PRINT@281,"Press <ENTER> to continue, M
for MENU";
10005 A$=INKEY$:IFA$="M"ORA$="m"
THENMAXFILES=0:CLEAR0:MENUELSEIFA$<>
CHR$(13)THEN10005ELSERETURN
```

Further compression (for example, adding line 205 to line 200) isn't practical since the final line would be too long to fit in BASIC, which has a maximum line length of about 240 characters.

As you can see from the above example, compressing a program makes it very hard to read and understand, so why would anyone want to do that? The reason is *room*. In the Model 100, space is at a premium. The less room a program uses, the more is available for other purposes. In BASIC programming, every time you eliminate a line number you save 4 bytes of memory. Thus, the above compressed program is actually 32 bytes smaller than the uncompressed program. In a machine like the Model 100, 32 bytes is an important amount.

Unfortunately, in the program listings the capital letter O looks like the number 0. Fortunately, only two programs, Invoice and Average Yield of an Investment, use capital O as a variable. For these two programs, refer to the variable cross-reference for the lines using that variable. In the rest of the programs, the capital O is used only in BASIC commands like GOTO and GOSUB. Everywhere else, you use the number 0.

Finally, one important note: if you do decide to modify these programs, don't use the variable A\$. It is already used in the line 10000-10005 subroutine. Since it is in virtually every program, we left it out of the cross-reference list.

1

Management Utilities

If a manager had a toolbox, what do you think he or she would carry in it, bullwhips and pink slips?

Well, perhaps, but far more valuable would be tools like the programs in this chapter.

The management utilities are highly useful, all-purpose tools intended for anyone in a decision-making position.

RATIOS

All businesses are constantly evaluated by themselves, banks, suppliers, investors, and others. A common method of evaluation used by these people is *ratio analysis*. By comparing relationships of different financial quantities within the business, they can estimate the overall health of the business. For further evaluation, the financial ratios of one business can be compared to the ratios of other businesses in the same industry.

This program asks for information about the company and calculates twelve common and useful ratios: current asset, quick ratio (acid test), debt to total assets, times interest earned, fixed charge coverage, inventory turnover, average collection period, fixed assets turnover, total assets turnover, profit margin on sales, return on total assets, and return on net worth. Once calculated, this information can be changed and the effect on the ratios viewed, yielding valuable information about the company and how it can best be managed.

In fact, we considered this program to be so important that we made it the first program in its chapter and moved the entire chapter to the front of the book (disturbing the overall sequential order of the programs).

An important feature of this program is its ability to manipulate the ratios to find how to make your business as strong as possible. For example, perhaps you can improve your current ratio by selling off inventory at a low price. With the Manipulate Ratio function, you choose the optimum value for the ratio and determine what the appropriate accounts should be. If you're unsure as to the meaning of any of these ratios, you'll find their descriptions in the Glossary at the end of the book.

When you first start the program, it asks for information about the company (sample answers are shown in bold type):

1—Current assets? **1000000**

Enter the value of your current assets, or enter 999 if you have previously saved your account totals with the save option of this program (described later).

2—Current liabilities? **300000**

3—Inventory? **1000000**

4—Total debt? **100000**

5—Total assets? **3000000**

- 6—Profit before taxes? 50000
- 7—Interest charges? 10000
- 8—Income available for meeting fixed charges? 300000
- 9—Fixed charges? 20000
- 10—Sales? 3000000
- 11—Fixed charges? 20000
- 12—Net profit after taxes? 65000
- 13—Net worth? 3000000

Answer each question with the value, if known. If the information is not available, simply press <ENTER>, and the ratio using that information will not be calculated.

14—Save data? Y

Answer Y or N. If you choose to save the data, a file will be opened called FINDAT.DO. The data can be loaded in again by answering the *Current assets?* prompt with the number 999.

The ratios will be calculated and printed:

- 1—Current asset 2.33 times
- 2—Quick ratio (acid test) -1.00 times
- 3—Debt to total assets 3.33 %
- 4—Times interest earned 6.00 times
- 5—Fixed charge coverage 15.00 times
- 6—Inventory turnover 3.00 times
- F1—Turn page F2—Manipulate

At this time, you may press the function key <F1> to see the rest of the ratios.

- 7—Avg. collection period 24.00 days
- 8—Fixed assets turnover 7.50 times
- 9—Total assets turnover 1.00 times
- 10—Profit margin on sales 2.17 %
- 11—Return on total assets 2.17 %
- 12—Return on net worth 2.17 %
- F1—Turn page F2—Manipulate

Press the function key <F2> to manipulate data.

- 7—Avg. collection period 24.00 days
- 8—Fixed assets turnover 7.50 times

9—Total assets turnover	1.00 times
10—Profit margin on sales	2.17 %
11—Return on total assets	2.17 %
12—Return on net worth	2.17 %

Change <D>ata or <R>atios?

This allows you either to change the original account data you entered or to change the ratios and have the corresponding data calculated for you. Answer the question with a D. You do not need to press <ENTER>. The screen will display:

1—Current assets = 700000
2—Current liabilities = 300000
3—Inventory = 1000000
4—Total debt = 100000
5—Total assets = 3000000
to change, x = exit, or <ENTER> = pag?
6—Profit before taxes = 50000
7—Interest charges = 10000
8—Income available for meeting fixed
charges = 300000
9—Fixed charges = 20000
10—Sales = 3000000
to change, x = exit, or <ENTER> = pag?
11—Receivables = 200000
12—Sales per day = 8333
13—Fixed assets = 400000
14—Net profit after taxes = 65000
15—Net worth = 3000000
to change, x = exit, or <ENTER> = pag?

To change the value of the net profit after taxes, enter the number 14 and press <ENTER>. The screen will display:

Net profit after taxes = 65000
Enter new value? **70000**

The ratios will be displayed again with new values for the ratios affected by the net profit.

7—Avg. collection period	24.00 days
8—Fixed assets turnover	7.50 times
9—Total assets turnover	1.00 times

10—Profit margin on sales	2.33 %
11—Return on total assets	2.33 %
12—Return on net worth	2.33 %
Change <D>ata or <R>atios?	

Press R. The screen will display:

7—Avg. collection period	24.00 days
8—Fixed assets turnover	7.50 times
9—Total assets turnover	1.00 times
10—Profit margin on sales	2.33 %
11—Return on total assets	2.33 %
12—Return on net worth	2.33 %
Enter = page, 0 = exit, or ratio number	

Enter the number 7 to change the average collection period.

Avg. collection days =	24.00
Function of:	
Receivables	200000
Sales	3000000
New ratio value?	30.00

The screen is showing the current value of the ratio and the accounts on which it is based. If, as in the above example, we changed the ratio value to 30.00, the program would calculate a new value for receivables and display:

Avg. collection days =	30.00
Function of:	
Receivables	250000
Sales	3000000
Change 11 = rcvbles, 10 = sales, 0 = exit? 11	

You can now alter the values of sales and receivables to the desired values. Each time you change one function, the other will automatically change so that the ratio remains accurate. When done, press 0 to go back to the ratio display mode. You will notice that, with many ratios, changing one causes others to change.

To make your company appear as strong as possible, find what changes need to be made to your accounts to cause the best possible ratio analysis of your company.

To find the ratios of other companies in the same industry, you

can calculate ratios from their annual reports or from researching ratio analysis studies published by such companies as Moody's and Standard and Poor's.

The file that is created with the save option will be called FINDAT.DO and will store only the values of each account in this manner:

700000
300000
1000000
100000
3000000
50000
10000
300000
20000
3000000
200000
8333
400000
65000
3000000

If the save option is used twice, the FINDAT.DO file will be erased and overwritten.

Line Number and Variable Cross-reference

Ref	Line Numbers				
+-----+					
:07370:	7360				
:07405:	7495				
:07440:	7500				
:07475:	7380	7400	7550		
:07485:	7490				
:07495:	7475	7480	7530	7605	7625
:	7670				
:07505:	7475				
:07515:	7515				
:07535:	7350				
:07555:	7520				
:07560:	7590				
:07585:	7570	7660	7665	7675	
:07595:	7525				
:07605:	7615	7630	7640		
:07645:	7580				
:07680:	7635				
:07685:	7735	7740	7745		
:07715:	7720				
:07735:	7710				
:07750:	7635				
:07755:	7810	7815	7820	7825	
:07790:	7795				
:07810:	7785				
:07830:	7635				
:07835:	7885	7890	7895		
:07865:	7870				
:07885:	7860				
:07900:	7635				
:07905:	7955	7960	7965		
:07935:	7940				
:07955:	7930				
:07970:	7635				
:07975:	8025	8030			
:08005:	8010				
:08025:	8000				
:08035:	7635				
:08040:	8090	8095	8100		
:08070:	8075				
:08090:	8065				
:08105:	7635				

:08110:	8160	8165	8170		
:08140:	8145				
:08160:	8135				
:08175:	7635				
:08180:	8230	8235	8240		
:08210:	8215				
:08230:	8205				
:08245:	7635				
:08250:	8300	8305	8310		
:08280:	8285				
:08300:	8275				
:08315:	7635				
:08320:	8370	8375	8380		
:08350:	8355				
:08370:	8345				
:08385:	7635				
:08390:	8440	8445	8450		
:08420:	8425				
:08440:	8415				
:08455:	7635				
:08460:	8510	8515	8520		
:08490:	8495				
:08510:	8485				
:A\$: 7305	7415	7440	7455	7460
:	7465	7690	7760	7840	7910
:	7980	8045	8115	8185	8255
:	8325	8395	8465		
:	PRINTUSING field format				
:B\$: 7305	7405	7410	7420	7425
:	7430	7445	7450		
:	PRINTUSING field format				
:I	: 7615	7625	7630	7635	
:	Keyboard input				
:I\$: 7375	7380	7515	7520	7525
:	7615	7650	7655	7660	7665
:	7670	7725	7730	7800	7805
:	7875	7880	7945	7950	8015
:	8020	8080	8085	8150	8155
:	8220	8225	8290	8295	8360
:	8365	8430	8435	8500	8505
:	Value of keyboard input				
:RA	: 7680	7690	7710	7735	7740
:	7750	7760	7785	7810	7815
:	7820	7830	7840	7860	7885
:	7890	7900	7910	7930	7955
:	7960	7970	7980	8000	8025

:		8030	8035	8045	8065	8090
:		8095	8105	8115	8135	8160
:		8165	8175	8185	8205	8230
:		8235	8245	8255	8275	8300
:		8305	8315	8325	8345	8370
:		8375	8385	8395	8415	8440
:		8445	8455	8465	8485	8510
:		8515				
:		Ratio value				
:S	:	7480	7495	7500	7530	7600
:		7625	7670			
:		Current Ratio	page			
:T	:	7680	7710	7750	7785	7830
:		7860	7900	7930	7970	8000
:		8035	8065	8105	8135	8175
:		8205	8245	8275	8315	8345
:		8385	8415	8455	8485	
:		Program flow	marker			
:X	:	7330	7355	7360	7365	7390
:		7540	7565	7570	7575	7580
:		7710	7715	7720	7725	7730
:		7735	7740	7785	7790	7795
:		7800	7805	7810	7815	7820
:		7860	7865	7870	7875	7880
:		7885	7890	7930	7935	7940
:		7945	7950	7955	7960	8000
:		8005	8010	8015	8020	8025
:		8030	8065	8070	8075	8080
:		8085	8090	8095	8135	8140
:		8145	8150	8155	8160	8165
:		8205	8210	8215	8220	8225
:		8230	8235	8275	8280	8285
:		8290	8295	8300	8305	8345
:		8350	8355	8360	8365	8370
:		8375	8415	8420	8425	8430
:		8435	8440	8445	8485	8490
:		8495	8500	8505	8510	8515
:		Loop counter	& Keyboard input			
:X\$(:	7315	7330	7345	7365	7570
:		7575	7665	7700	7705	7725
:		7770	7775	7780	7800	7850
:		7855	7875	7920	7925	7945
:		7995	8015	8055	8060	8080
:		8125	8130	8150	8195	8200
:		8220	8265	8270	8290	8335
:		8340	8360	8405	8410	8430

10 60 Business Applications Programs / TRS-80 Model 100

```
      :      8475      8480      8500
      :      Account names
:X(   : 7315      7345      7350      7360      7365
      :      7390      7405      7410      7415      7420
      :      7425      7430      7440      7445      7450
      :      7455      7460      7465      7540      7570
      :      7575      7665      7680      7700      7705
      :      7730      7735      7740      7750      7770
      :      7775      7780      7805      7810      7815
      :      7820      7830      7850      7855      7880
      :      7885      7890      7900      7920      7925
      :      7950      7955      7960      7970      7990
      :      7995      8020      8025      8030      8035
      :      8055      8060      8085      8090      8095
      :      8105      8125      8130      8155      8160
      :      8165      8175      8195      8200      8225
      :      8230      8235      8245      8265      8270
      :      8295      8300      8305      8315      8335
      :      8340      8365      8370      8375      8385
      :      8405      8410      8435      8440      8445
      :      8455      8475      8480      8505      8510
      :      8515
      :      Account data
+-----+
```

Ratios Program

```

7300 CLS:MAXFILES=1:CLEAR500:PRINTTAB(15
)"RATIO ANALYSIS":REM RATIO
7305 A$="####.##":B$="####.##"
7310 KEY ON
7315 DIM X(16),X$(16)
7320 DATA"Current assets","Current liabi
lities","Inventory","Total debt","Total
assets","Profit before taxes","Interest
charges"
7325 DATA"Income available for meeting f
ixed charges","Fixed charges","Sale
s","Recievables","Sales per day","Fixed
assets","Net profit after taxes","Net wo
rth"
7330 FORX=1TO15:READX$(X):NEXT
7335 PRINT"This program will calculate r
atios for given information and allow m
anipulationof the information."
7340 PRINT"Should any information be una
vailable, just press <ENTER>."
7345 PRINTX$(1);:INPUTX(1)
7350 IFX(1)=999 THEN GOTO 7535
7355 FORX=2TO15
7360 IF X=12 THEN X(X)=INT(X(10)/360):GO
TO7370
7365 PRINTX$(X);:INPUTX(X)
7370 NEXT
7375 INPUT"Save data";I$
7380 IF I$<>"Y"ANDI$<>"y"THEN7475
7385 OPEN"RAM:FINDAT.DO"FOR OUTPUT AS 1
7390 FORX=1TO15:PRINT#1,X(X):NEXT
7395 CLOSE
7400 GOTO 7475
7405 CLS:IFX(1)<>0 AND X(2)<>0 THEN PRIN
T"1-Current asset ";:PRINTUSIN
GB$;X(1)/X(2);:PRINT" times"
7410 IFX(1)<>0 AND X(2)<>0 AND X(3)<>0 T
HEN PRINT"2-Quick ratio (acid test) ";:P
RINTUSINGB$;(X(1)-X(3))/X(2);:PRINT" tim
es"
7415 IFX(4)<>0 AND X(5)<>0 THEN PRINT"3-
Debt to total assets ";:PRINTUSINGA$;
(X(4)/X(5))*100;:PRINT" %"

```



```
7420 IF X(6)<>0ANDX(7)<>0THENPRINT"4-Tim
es interest earned ";:PRINTUSINGB$(X(
6)+X(7))/X(7);:PRINT" times"
7425 IF X(9)<>0 AND X(8)<>0 THEN PRINT"5
-Fixed charge coverage ";:PRINTUSINGB$
;X(8)/X(9);:PRINT" times"
7430 IF X(10)<>0ANDX(3)<>0THENPRINT"6-In
ventory turnover ";:PRINTUSINGB$;X(
10)/X(3);:PRINT" times"
7435 RETURN
7440 CLS:IF X(11)<>0ANDX(10)<>0THENPRINT
"7 -Avg. collection period ";:PRINTUSING
A$;X(11)/INT(X(10)/360);:PRINT" days"
7445 IFX(10)<>0ANDX(13)<>0THENPRINT"8 -F
ixed assets turnover ";:PRINTUSINGB$;X(
10)/X(13);:PRINT" times"
7450 IFX(10)<>0ANDX(5)<>0THENPRINT"9 -To
tal assets turnover ";:PRINTUSINGB$;X(1
0)/X(5);:PRINT" times"
7455 IFX(14)<>0ANDX(10)<>0THENPRINT"10-P
rofit margin on sales ";:PRINTUSINGA$;(X
(14)/X(10))*100;:PRINT" %"
7460 IFX(14)<>0ANDX(5)<>0THENPRINT"11-Re
turn on total assets ";:PRINTUSINGA$;(X(
14)/X(5))*100;:PRINT" %"
7465 IFX(14)<>0ANDX(15)<>0THENPRINT"12-R
eturn on net worth ";:PRINTUSINGA$;(X
(14)/X(15))*100;:PRINT" %"
7470 RETURN
7475 KEY ON:ON KEY GOSUB 7495,7505
7480 S=0:GOSUB 7495
7485 PRINT@280,"F1-Turn page F2-manipul
ate ";
7490 GOTO 7485
7495 IF S=0 THEN S=1: GOSUB 7405:RETURN
7500 IF S=1 THEN S=0:GOSUB 7440:RETURN
7505 KEY (2) OFF
7510 PRINT@280,"Change <D>ata or <R>atio
s? ";
7515 I$=INKEY$:IFI$=""THEN7515
7520 IF I$="D"OR I$="d"THENGOTO7555
7525 IFI$="R"OR I$="r"THEN GOTO 7595
7530 KEY ON:S=0:GOSUB 7495:RETURN
7535 OPEN"RAM:FINDAT.DO"FOR INPUT AS 1
7540 FORX=1TO15:INPUT#1,X(X):NEXT
7545 CLOSE
```

```
7550 GOTO 7475
7555 KEY(1)OFF
7560 CLS
7565 FORX=1TO15
7570 IFX=12 THENPRINTX;"-";X$(X);"=";INT
(X(10)/360):GOTO7585
7575 PRINTX;"-";X$(X);"=";X(X)
7580 IFX/5=INT(X/5)THENGOTO7645:CLS
7585 NEXT
7590 GOTO7560
7595 KEY OFF
7600 S=0
7605 CLS:GOSUB 7495
7610 PRINT@280,"Enter=page, 0=exit, or r
atio number ";
7615 INPUTI$:I=VAL(I$):IFI$=CHR$(114)THE
NGOTO7605
7620 CLS
7625 IF I=0 THEN KEY ON:S=0:GOSUB7495:RE
TURN
7630 IFI<1 OR I>12 THEN 7605
7635 ON I GOSUB 7680,7750,7830,7900,7970
,8035,8105,8175,8245,8315,8385,8455
7640 GOTO 7605
7645 PRINT@280,"# to change, x=exit, or
<ENTER>=page";
7650 I$=""
7655 PRINT@315,"";:INPUTI$:CLS:
7660 IF VAL(I$)=12 THEN CLS:PRINT:PRINT"
Sales per day is a function of total
sales.":PRINT@280,"Press <ENTER> to cont
inue";:INPUTI$:GOTO7585
7665 IFVAL(I$)>0 ANDVAL(I$)<16THENPRINT:
PRINTX$(VAL(I$));"=";X(VAL(I$)):PRINT@28
0,"Enter new value
";:PRINT@298,"";:INPUTX(VAL(I$)):CLS:G
OTO7585
7670 IFI$="x"ORIS$="X"THEN KEY ON:S=0:GOS
UB7495:RETURN
7675 CLS:GOTO7585
7680 T=0:RA=X(1)/X(2)
7685 CLS
7690 PRINT"Current asset="";:PRINT@28,"";
:PRINTUSINGA$;RA
7695 PRINT:PRINT"function of:"
7700 PRINTX$(1);:PRINT@148,X(1)
```



```
7705 PRINTX$(2);:PRINT@188,X(2)
7710 IFT=0 THEN T=1:PRINT@240,"New ratio
      value";:INPUT RA:X=1:GOTO7735
7715 PRINT@240,"Change 1=Assets 2=Liabil
      ities 0=Exit";:INPUTX:IFX=0THEN RETURN
7720 IF X>2THEN7715
7725 PRINT@280,"New value for ";:PRINTX$
      (X);:INPUTI$:
7730 X(X)=VAL(I$)
7735 IFX=1THEN X(2)=INT(X(1)/RA):GOTO 76
      85
7740 IFX=2THEN X(1)=INT(X(2)*RA):GOTO768
      5
7745 GOTO 7685
7750 T=0:RA=(X(1)-X(3))/X(2)
7755 CLS
7760 PRINT"Quick ratio (acid)=";:PRINT@2
      8,"";:PRINTUSINGA$;RA
7765 PRINT:PRINT"function of:"
7770 PRINTX$(1);:PRINT@148,X(1)
7775 PRINTX$(2);:PRINT@188,X(2)
7780 PRINTX$(3);:PRINT@228,X(3)
7785 IFT=0 THEN T=1:PRINT@240,"New ratio
      value";:INPUT RA:X=1:GOTO7810
7790 PRINT@240,"Change 1=Ast. 2=Liab. 3=
      Inv. 0=Exit";:INPUTX:IFX=0THENRETURN
7795 IFX>3THEN7790
7800 PRINT@280,"New value for ";:PRINTX$
      (X);:INPUTI$:
7805 X(X)=VAL(I$)
7810 IFX=1THENX(2)=INT((X(1)-X(3))/RA):G
      OTO 7755
7815 IFX=2THENX(3)=INT(X(1)-(RA*X(2))):G
      OT07755
7820 IFX=3THENX(1)=INT((RA*X(2))+X(3)):G
      OT07755
7825 GOTO7755
7830 T=0:RA=X(4)/X(5)
7835 CLS
7840 PRINT"Debt to total assets=";:PRINT
      @28,"";:PRINTUSINGA$;RA*100;:PRINT"%"
7845 PRINT:PRINT"function of:"
7850 PRINTX$(4);:PRINT@148,X(4)
7855 PRINTX$(5);:PRINT@188,X(5)
7860 IFT=0 THEN T=1:PRINT@240,"New ratio
```



```
value %";:INPUT RA:RA=RA/100:X=4:GOTO78
85
7865 PRINT@240,"Change 4=Debt 5=Assets 0
=Exit";:INPUTX:IFX=0THEN RETURN
7870 IFX<>4ANDX<>5THEN7865
7875 PRINT@280,"New value for ";:PRINTX$
(X);:INPUTI$:
7880 X(X)=VAL(I$)
7885 IFX=4THEN X(5)=INT(X(4)/RA):GOTO 78
35
7890 IFX=5THEN X(4)=INT(X(5)*RA):GOTO783
5
7895 GOTO 7835
7900 T=0:RA=(X(6)+X(7))/X(7)
7905 CLS
7910 PRINT"Times interest earned=";:PRIN
T@28,"";:PRINTUSINGA$;RA
7915 PRINT:PRINT"function of:"
7920 PRINTX$(6);:PRINT@148,X(6)
7925 PRINTX$(7);:PRINT@188,X(7)
7930 IFT=0 THEN T=1:PRINT@240,"New ratio
value";:INPUT RA:X=6:GOTO7955
7935 PRINT@240,"Change 6=Profit 7=Intere
st 0=Exit";:INPUTX:IFX=0THEN RETURN
7940 IF X<6ANDX>7THEN7935
7945 PRINT@280,"New value for ";:PRINTX$
(X);:INPUTI$:
7950 X(X)=VAL(I$)
7955 IFX=6THEN X(7)=INT(X(6)/(RA-1)):GOT
O 7905
7960 IFX=7THEN X(6)=INT(X(7)*(RA-1)):GOT
O7905
7965 GOTO 7905
7970 T=0:RA=X(8)/X(9)
7975 CLS
7980 PRINT"Fixed charge coverage=";:PRIN
T@28,"";:PRINTUSINGA$;RA
7985 PRINT:PRINT"function of:"
7990 PRINT"Income available";:PRINT@148,
X(8)
7995 PRINTX$(9);:PRINT@188,X(9)
8000 IFT=0 THEN T=1:PRINT@240,"New ratio
value";:INPUT RA:X=8:GOTO8025
8005 PRINT@240,"Change 8=Profit 9=Intere
st 0=Exit";:INPUTX:IFX=0THEN RETURN
```



```
8010 IF X<8ANDX>9THEN8005
8015 PRINT@280,"New value for ";;PRINTLE
FT$(X$(X),17);:INPUTI$:
8020 X(X)=VAL(I$)
8025 IFX=8THEN X(9)=INT(X(8)/RA):GOTO 79
75
8030 IFX=9THEN X(8)=INT(RA*X(9)):GOTO797
5
8035 T=0:RA=X(10)/X(3)
8040 CLS
8045 PRINT"Inventory turnover=";;PRINT@2
8,"";:PRINTUSINGA$;RA
8050 PRINT:PRINT"function of:"
8055 PRINTX$(10);:PRINT@148,X(10)
8060 PRINTX$(3);:PRINT@188,X(3)
8065 IFT=0 THEN T=1:PRINT@240,"New ratio
value";:INPUT RA:X=10:GOTO8090
8070 PRINT@240,"Change 10=Sales 3=Invent
ory 0=Exit";:INPUTX:IFX=0THEN RETURN
8075 IF X<>3ANDX<>10THEN8070
8080 PRINT@280,"New value for ";;PRINTX$
(X);:INPUTI$:
8085 X(X)=VAL(I$)
8090 IFX=10THEN X(3)=INT(X(10)/RA):GOTO
8040
8095 IFX=3THEN X(10)=INT(X(3)*RA):GOTO80
40
8100 GOTO 8040
8105 T=0:RA=INT(X(11)/(X(10)/360))
8110 CLS
8115 PRINT"Avg. Collection days=";;PRINT
@28,"";:PRINTUSINGA$;RA
8120 PRINT:PRINT"function of:"
8125 PRINTX$(11);:PRINT@148,X(11)
8130 PRINTX$(10);:PRINT@188,X(10)
8135 IFT=0 THEN T=1:PRINT@240,"New ratio
value";:INPUT RA:X=10:GOTO8160
8140 PRINT@240,"Change 11=Rcvbles. 10=Sa
les 0=Exit";:INPUTX:IFX=0THEN RETURN
8145 IF X<>11ANDX<>10THEN8140
8150 PRINT@280,"New value for ";;PRINTX$
(X);:INPUTI$:
8155 X(X)=VAL(I$)
8160 IFX=11THEN X(10)=INT((X(11)*360)/RA
):GOTO 8110
```



```
8165 IFX=10THEN X(11)=INT((X(10)*RA)/360
):GOTO8110
8170 GOTO 8110
8175 T=0:RA=X(10)/X(13)
8180 CLS
8185 PRINT"Fixed asset turnover=";:PRINT
@25,"";:PRINTUSINGA$;RA;:PRINT" times"
8190 PRINT:PRINT"function of:"
8195 PRINTX$(10);:PRINT@148,X(10)
8200 PRINTX$(13);:PRINT@188,X(13)
8205 IFT=0 THEN T=1:PRINT@240,"New ratio
value %";:INPUT RA:X=10:GOTO8230
8210 PRINT@240,"Change 10=Sales 13=Asset
s 0=Exit";:INPUTX:IFX=0THEN RETURN
8215 IFX<>10ANDX<>13THEN8210
8220 PRINT@280,"New value for ";:PRINTX$
(X);:INPUTI$:
8225 X(X)=VAL(I$)
8230 IFX=10THEN X(13)=INT(X(10)/RA):GOTO
8180
8235 IFX=13THEN X(10)=INT(X(13)*RA):GOTO
8180
8240 GOTO 8180
8245 T=0:RA=X(10)/X(5)
8250 CLS
8255 PRINT"Total asset turnover=";:PRINT
@25,"";:PRINTUSINGA$;RA:PRINT" times"
8260 PRINT:PRINT"function of:"
8265 PRINTX$(10);:PRINT@148,X(10)
8270 PRINTX$(5);:PRINT@188,X(5)
8275 IFT=0 THEN T=1:PRINT@240,"New ratio
value %";:INPUT RA:X=10:GOTO8300
8280 PRINT@240,"Change 10=Debt 5=Assets
0=Exit";:INPUTX:IFX=0THEN RETURN
8285 IFX<>10ANDX<>5THEN8280
8290 PRINT@280,"New value for ";:PRINTX$
(X);:INPUTI$:
8295 X(X)=VAL(I$)
8300 IFX=10THEN X(5)=INT(X(10)/RA):GOTO
8250
8305 IFX=5THEN X(10)=INT(X(5)*RA):GOTO82
50
8310 GOTO 8250
8315 T=0:RA=X(14)/X(10)
8320 CLS
```



```
8325 PRINT"Profit margin on sales=";:PRI
NT@27,"";:PRINTUSINGA$;RA*100;:PRINT" %"
8330 PRINT:PRINT"function of:"
8335 PRINTX$(14);:PRINT@148,X(14)
8340 PRINTX$(10);:PRINT@188,X(10)
8345 IFT=0 THEN T=1:PRINT@240,"New ratio
value %";:INPUT RA:RA=RA/100:X=14:GOTO8
370
8350 PRINT@240,"Change 14=Profit 10=Sale
s 0=Exit";:INPUTX:IFX=0THEN RETURN
8355 IFX<>14ANDX<>10THEN8350
8360 PRINT@280,"New value for ";:PRINTX$
(X);:INPUTI$:
8365 X(X)=VAL(I$)
8370 IFX=14THENX(10)=INT(X(14)/RA):GOTO
8320
8375 IFX=10THENX(14)=INT(X(10)*RA):GOTO
8320
8380 GOTO 8320
8385 T=0:RA=X(14)/X(5)
8390 CLS
8395 PRINT"Return on total assets=";:PRI
NT@28,"";:PRINTUSINGA$;RA*100;:PRINT"%"
8400 PRINT:PRINT"function of:"
8405 PRINTX$(14);:PRINT@148,X(14)
8410 PRINTX$(5);:PRINT@188,X(5)
8415 IFT=0 THEN T=1:PRINT@240,"New ratio
value %";:INPUT RA:RA=RA/100:X=14:GOTO8
440
8420 PRINT@240,"Change 14=Profit 5=Asset
s 0=Exit";:INPUTX:IFX=0THEN RETURN
8425 IFX<>14ANDX<>5THEN8420
8430 PRINT@280,"New value for ";:PRINTX$
(X);:INPUTI$:
8435 X(X)=VAL(I$)
8440 IFX=14THENX(5)=INT(X(14)/RA):GOTO 8
390
8445 IFX=5THENX(14)=INT(X(5)*RA):GOTO 83
90
8450 GOTO 8390
8455 T=0:RA=X(14)/X(15)
8460 CLS
8465 PRINT"Return on net worth=";:PRINT@
28,"";:PRINTUSINGA$;RA*100;:PRINT"%"
8470 PRINT:PRINT"function of:"
8475 PRINTX$(14);:PRINT@148,X(14)
8480 PRINTX$(15);:PRINT@188,X(15)
```



```
8485 IFT=0 THEN T=1:PRINT@240,"New ratio  
value %";:INPUT RA:RA=RA/100:X=14:GOTO8  
510  
8490 PRINT@240,"Change 14=Profit 15=Net  
worths 0=Exit";:INPUTX:IFX=0THEN RETURN  
8495 IFX<>14ANDX<>15THEN8490  
8500 PRINT@280,"New value for ";:PRINTX$  
(X);:INPUTI$:  
8505 X(X)=VAL(I$)  
8510 IFX=14THENX(15)=INT(X(14)/RA):GOTO  
8460  
8515 IFX=15THENX(14)=INT(X(15)*RA):GOTO  
8460  
8520 GOTO 8460
```

CALCULATOR

This program is a useful, flexible, business-oriented calculator. Its primary function is data storage in seven namable memories that are always in view. Adding up seven columns of figures in a spread sheet becomes simple with this program.

There are two types of calculators: *linear* and *algebraic*. Linear calculators expect entries like this:

123 - 23

with the answer printed immediately afterward:

100

Algebraic calculators expect entries like this:

123+
23-

Again, the answer, of course, is 100.

This calculator works either way. The Calculator program starts up ready to go: simply begin putting in your figures however you please.

There are four significant sections on the calculator screen:

1. The *box*. In the upper left-hand side of the screen is a 15-character display. This will hold the current totals.
 2. Directly to the left of the box is a *space for the mathematical operator currently in use*. It will hold the signs +, -, ×, and / during addition, subtraction, multiplication, and division, respectively.
 3. The *entry line* is in the lower left-hand part of the display. It is preceded by the symbols = >. The numbers you enter will appear here.
 4. The *memory storage section* is on the right-hand side of the screen. The memory modules, initially named M0, M1, M2, M3, M4, M5, and M6, and their contents are displayed.
-

To put a number in the box, either press one of the arithmetic operators (*, ×, +, -, =, /, r, or <GRPH>r), or press <ENTER>.

Press the LABEL function key. The bottom line of the screen will display:

Stor Rcll Sum Pntr Clr ClrM MNam Quit

These are your choices for manipulating the numbers you're typing into the calculator. Here's what the various functions are:

Stor (Store) This is used to store the contents of the box to memory. When F1 is pressed, the entry line will display:

Stor #?

Enter the number of the memory module in which you would like the data stored. The designated memory number will now reflect the new data.

Rcll (Recall) This will bring data from the memory to the box for calculation. When the F2 key is pressed, the entry line will display:

Rcll #?

Press the number of the memory module that you wish to recall.

Sum This will add the number in the box to the specified memory number. When the F3 key is pressed, the entry line will display:

Sum #?

Enter the number of the memory to which the number in the box is to be added.

Pntr (Printer) When the F4 key is pressed, all entries will be printed out on your printer, if it's connected and turned on. Pressing F4 again will cause the printing to stop. If the printer is off or not connected, the program will freeze until the printer is on and connected.

Clr (Clear) When the F5 key is pressed, the box will clear, and a zero will be displayed.

ClrM (Clear Memory) When the F6 key is pressed, all the memories will be cleared and their values set to zero. All data is lost. The value in the box, if any, will not be affected.

MNam (Memory Name) This allows you to rename the memories. This can be used to remind you what information you are storing in each of the seven memories. When the F7 key is pressed, the entry line will display:

MNam #?

Enter the memory number you wish changed. The entry line will display:

New Name?

Enter the name that you want to appear instead of the M0-M6 labels that appear on initial program use. Only the first 5 characters of the name you enter will be used, and the new memory name will be preceded by the memory number. For example, if you change the name of memory module 2 to *Expenses*, then the name will appear as *2Expens*.

Quit When the F8 key is pressed, the entry line will display the question:

Save totals and memory names (Y/N)?

There is no need to press <ENTER> after pressing a Y or N. If you answer Y, the entry line will display:

Filename?

Answer the question. If your answer is more than 6 characters, any additional characters will be ignored. A file will then be created that contains the amount in the box, the memory names, and the amounts stored in the memories. The entry line will display:

Quit (Y/N)?

If you are not done, answer the question with N; otherwise, your menu will be displayed. When you Quit, your function keys will be restored to those with which the computer was originally delivered, not those you last used.

Calculations

The acceptable calculation operators are:

* or ×	Multiplication
/	Division
^	Exponentiation (<SHIFT> 6)
<GRPH>r	Square root
+	Addition
-	Subtraction
= or <ENTER>	Equals

Line Number and Variable Cross-reference

Ref	Line Numbers				
+-----+					
:05310:	5310				
:05370:	5490				
:05450:	5525	5555	5730		
:05490:	5360	5620	5640	5660	5670
:	5750	5780			
:05495:	5525	5530	5555	5730	5800
:05500:	5610				
:05505:	5505	5520			
:05560:	5555				
:05565:	5555				
:05570:	5555				
:05575:	5555				
:05580:	5555				
:05585:	5555				
:05590:	5555				
:05595:	5535				
:05605:	5600				
:05625:	5600	5630			
:05645:	5600	5650			
:05665:	5600				
:05675:	5785				
:05680:	5680				
:05730:	5600				
:05735:	5600				
:05755:	5600	5760			
:05785:	5600				
:05790:	5790				
:05800:	5540				
:05805:	5795				
:05810:	5340				
:05820:	5825				
:05830:	5820				
:B	5810	5830			
:	Function key decode operator				
:E	5810	5820	5830		
:	Function key decode operator				
:F	5530	5550	5555		
:	Operator type				
:F1\$	5325	5520	5550		


```

:      acceptable calculation symbols.
:GT   : 5450   5475   5525   5560   5565
:      : 5570   5575   5580   5585   5615
:      : 5635   5655   5720   5730
:      'Box' value.
:GT$  : 5450   5455   5460   5465   5480
:      The fifteen most significant
:      digits of the 'box' value.
:I$   : 5310   5315   5505   5520   5530
:      : 5555   5680   5685
:      Contains the character entered.
:K$   : 5335   5535   5805   5820   5830
:      Holds the function key labels.
:L    : 5480   5550   5665
:      Lineprint flag
:L$   : 5830
:      Function key declaration
:M$   : 5330   5350
:      Holds the original memory names
:M$(  : 5350   5410   5415   5710   5775
:      Memory names
:M(   : 5425   5615   5635   5655   5710
:      : 5740
:      Memory values
:MT$  : 5425   5430   5435   5440
:      Nine most significant
:      digits of the memory value.
:S    : 5315   5510   5790
:      =1 for sound, =0 for none
:TI$  : 5500   5520   5530   5535   5545
:      : 5605   5625   5645   5755
:      Contains entry line input
:TT   : 5545   5550   5560   5565   5570
:      : 5580   5585
:      Value entered on the entry line.
:U    : 5515   5520   5540
:      = 1 if user just entered a cal-
:      culation symbol.
:X    : 5345   5350   5390   5395   5400
:      : 5405   5410   5415   5425   5520
:      : 5525   5530   5555   5595   5600
:      : 5605   5610   5615   5625   5630
:      : 5635   5645   5650   5655   5705
:      : 5710   5735   5740   5755   5760
:      : 5775   5815

```



```
      :      calculation symbol position
:X$   : 5690   5695   5700   5765   5770
:     : 5775   5790   5795
:     : Filename
:X1   : 5535   5595
:     : Miscellaneous variable
+-----+
```


Calculator Program

```
5300 ' calculator
5305 CLS:PRINT@175,"Sound (Y/N)";
5310 I$=INKEY$:IFI$=""THEN5310
5315 IFI$="Y"OR I$="y"THENS=1:SOUND16383,
5:ELSES=0
5320 CLS
5325 F1$="*x/^(CHR$(137)+"+-="+CHR$(13)
5330 M$="MOM1M2M3M4M5M6"
5335 K$="Stor #/Rcll #/Sum # /Pntr "+CHR
$(13)+"/Clr "+CHR$(13)+"/ClrM "+CHR$(13
)+"/MNam #/Quit "+CHR$(13)+"/"
5340 GOSUB5810
5345 FORX=0TO6
5350 M$(X)=MID$(M$,(X*2)+1,2)
5355 NEXT
5360 GOTO5490
5365 END
5370 CLS
5375 PRINT@1,CHR$(240);STRING$(17,CHR$(2
41));CHR$(242);
5380 PRINT@41,CHR$(245);STRING$(17,CHR$(
32));CHR$(245);
5385 PRINT@81,CHR$(246);STRING$(17,CHR$(
241));CHR$(247)
5390 FORX=1TO3
5395 NEXTX
5400 FORX=0TO6
5405 PRINT@((X*40)+20),CHR$(245);
5410 IFLEN(M$(X))<6 THEN M$(X)=M$(X)+STR
ING$(6-LEN(M$(X))," ")
5415 PRINTM$(X);
5420 PRINT"=";
5425 MT$=STR$(M(X))
5430 IFLEN(MT$)>9THENMT$=LEFT$(MT$,9)
5435 IFLEN(MT$)<9THENMT$=STRING$(9-LEN(M
T$)," ")+MT$
5440 PRINTMT$;
5445 NEXT
5450 GT$=STR$(GT)
5455 IFLEN(GT$)>15THENG$=LEFT$(GT$,15)
5460 IFLEN(GT$)<15THENG$=STRING$(15-LEN
(GT$)," ")+GT$
5465 PRINT@42,GT$;
5470 PRINT@40," ";
```



```
5475 IFGT>9999999999999999THENPRINT@122,"0
verflow";:ELSEPRINT@122,"          ";
5480 IFL=1THENLPRINT"=";GT$
5485 RETURN
5490 GOSUB5370
5495 PRINT@240,"=>          ";;PRINT
@242,"";
5500 TI$=""
5505 I$=INKEY$:IFI$=""THEN 5505
5510 IFS=1THENSOUND16383,5
5515 U=0
5520 X=INSTR(1,F1$,I$):IFX=0THENPRINTI$;
:TI$=TI$+I$:GOTO5505ELSEU=1
5525 IFX=5THENGTSQR(GT):GOSUB5450:GOTO5
495
5530 IF LEN(TI$)=0THENPRINT@40,I$:F=X:GO
TO5495
5535 X1=INSTR(1,K$,LEFT$(TI$,4)):IFX1>0T
HEN5595
5540 IFU=0THEN5800
5545 TT=VAL(TI$)
5550 IFF>0THENIFL=1THENLPRINTMID$(F1$,F,
1);TT
5555 ONF+1 GOSUB5580,5560,5560,5565,5570
,5575,5580,5585,5590,5590:GOSUB5450:F=X:
PRINT@40,I$;;GOTO5495
5560 GT=GT*TT:RETURN
5565 GT=GT/TT:RETURN
5570 GT=GT^TT:RETURN
5575 GT=SQR(GT):RETURN
5580 GT=GT+TT:RETURN
5585 GT=GT-TT:RETURN
5590 RETURN
5595 X=((X1-1)/7)+1
5600 ON X GOTO 5605,5625,5645,5665,5730,
5735,5755,5785
5605 X=VAL(RIGHT$(TI$,1))
5610 IFX<0ORX>7THENBEEP:GOTO5500
5615 M(X)=GT
5620 GOTO5490
5625 X=VAL(RIGHT$(TI$,1))
5630 IFX<0ORX>7THENBEEP:GOTO5625
5635 GT=M(X)
5640 GOTO5490
5645 X=VAL(RIGHT$(TI$,1))
5650 IFX<0ORX>6THENBEEP:GOTO5645
```



```
5655 M(X)=M(X)+GT
5660 GOTO5490
5665 IFL=1THENL=OELSEL=1
5670 GOTO5490
5675 CLS:PRINT"Save totals and memory na
mes (Y/N)"
5680 I$=INKEY$:IFI$=""THEN5680
5685 IFI$="n"ORIS="N"THENRETURN
5690 PRINT@240,"Filename";:INPUTX$
5695 IFLN(X$)>6THENX$=LEFT$(X$,6)
5700 OPEN"RAM:"+X$+".do"FOROUTPUTAS1
5705 FORX=0TO6
5710 PRINT#1,M$(X);"=";M(X)
5715 NEXT
5720 PRINT#1,"Display = ";GT
5725 RETURN
5730 GT=0:GOSUB5450:GOTO5495
5735 FORX=0TO6
5740 M(X)=0
5745 NEXT
5750 GOTO5490
5755 X=VAL(RIGHT$(TI$,1))
5760 IFX>6ORX<0THENBEEP:GOTO5755
5765 PRINT@240,"New name";:INPUTX$
5770 IFLN(X$)>5THENX$=LEFT$(X$,5)
5775 M$(X)=STR$(X)+X$:M$(X)=RIGHT$(M$(X)
,LEN(M$(X))-1)
5780 GOTO 5490
5785 GOSUB5675:CLS:PRINT@240,"Quit (Y/N)
";
5790 X$=INKEY$:IFX$=""THEN5790ELSEIFS=1T
HENSOUND16383,5
5795 IFX$="Y"ORX$="y"THENGOSUB5805:MENU
5800 PRINT@240,"unknown";:BEEP:GOTO5495
5805 K$="Files"+CHR$(13)+"/Load "+CHR$(3
4)+"/Save/Run"+CHR$(13)+"/List"+CHR$(13)
+"///Menu"+CHR$(13)+"/"
5810 B=1:E=0
5815 FORX=1TO8
5820 E=E+1:IFMID$(K$,E,1)="/"THEN5830
5825 GOTO5820
5830 L$=MID$(K$,B,E-B):B=E+1:KEYX,L$:NEX
T
5835 RETURN
```


BREAKEVEN UNIT ANALYSIS

This business analysis utility can be invaluable in helping you evaluate the profitability of products and in determining optimum levels of pricing, sales, and costs.

When you first run the program, you are asked for information about your product and your company. If you press <ENTER> without typing a number first, a zero is automatically used as the value. After you have entered all the fixed and variable costs (plant and equipment depreciation, rentals, interest rate on debts, administrative staff salaries, general office expenses, miscellaneous fixed expenses, labor costs per unit, material costs per unit, sales commissions per unit, other miscellaneous costs per unit, and, finally, the sales price per unit), the program lists the totals of the fixed costs, variable costs, and sales price.

Underneath these choices is listed *Breakeven Units*, the number of units you need to sell in order to have a net profit of zero. The program then lets you change these factors or continue to part B of the program. As you change the information in items 1 (fixed costs), 2 (variable costs), and 3 (sales price), the number of breakeven units will change. Continue making these changes until you are satisfied with the data. If there is no breakeven point, then the word *Error* will appear on the *Breakeven Units* line.

When you've finished with part A, enter 0. You're now in part B of the program, the profit analysis portion. As in part A, you can change the information in the four categories (fixed costs, variable costs, sales price, and units sold).

Example

<i>Program</i>	<i>Your response</i>
Enter fixed costs <ENTER> if none:	
Depreciation—plant & equip.?	1000
Rentals?	100
Interest charges on debt?	50
Salaries for admin. staff?	275
General office expenses?	100
Other fixed costs?	45

Enter variable costs PER UNIT	
Factory labor?	1.12
Materials?	2.02
Sales commissions?	5.00
Other variable costs?	
Sales price per unit?	50.00
0—Calculate profit	
1—Fixed costs \$1570	
2—Variable costs \$8.14	
3—Sales price \$50	
Breakeven Units 37	
Change # or 0?	3
Enter new value?	75
0—Calculate profit	
1—Fixed costs \$1570	
2—Variable costs \$8.14	
3—Sales price \$75	
Breakeven Units 23	
Change # or 0?	2
Enter new value?	1600
0—Calculate profit	
1—Fixed costs \$1570	
2—Variable costs \$1600	
3—Sales price \$75	
Breakeven Units Error	
Change # or 0?	2
Enter new value?	50
0—Calculate profit	
1—Fixed costs \$1700	
2—Variable costs \$50	
3—Sales price \$100	
Breakeven Units 34	
Change # or 0?	0
0—End	
1—Fixed costs \$1700	
2—Variable costs \$50	
3—Sales price \$100	
4—Units sold 34	
Profit (- = loss) \$0	
Change # or 0?	4
Enter new value?	1000

0—End

1—Fixed costs \$1700

2—Variable costs \$50

3—Sales price \$100

4—Units sold 1000

Profit (- = loss) \$48300

Change # or 0?

0

Line Number and Variable Cross-reference

Ref	Line Numbers				
+-----+					
:06920:	6935				
:06940:	6920				
:06945:	6960				
:06965:	6950				
:06970:	7040				
:06980:	7045				
:07010:	7015				
:07045:	7020	7040			
:FC	: 6925	6930			
:	Fixed cost				
:I\$: 6920	6930	6950	6955	
:	Company data				
:VC	: 6945	6955			
:	Variable costs				
:X	: 7010	7015	7020	7025	7035
:	Keyboard input				
:X\$(: 6975	7005			
:	Breakeven units				
:X(: 6930	6955	6965	6970	6975
:	6990	6995	7000	7005	7035
:	7045				
:	Data				
:Y	: 7030	7035			
:	Keyboard input				
:Z	: 6985	7005	7015	7020	7040
:	7045				
:	Part A or B				
+-----+					

Breakeven Unit Analysis Program

```
6900 'Breakeven units analysis      breaku
6905 DATA "Depreciation - plant & equip.
      ","Rentals","Interest charges on debt","
      Salaries for admin staff","General offic
      e expenses","Other fixed costs","x"
6910 DATA "Factory labor","Materials","s
      ales commissions","Other variable costs"
      ,"x"
6915 CLS:PRINT"Enter fixed costs <ENTER>
      if none:"
6920 READI$:IFI$="x"THEN6940
6925 FC=0
6930 PRINTI$;:INPUTFC:X(1)=X(1)+FC
6935 GOTO6920
6940 CLS:PRINT"Enter variable costs PER
      UNIT"
6945 VC=0
6950 READI$:IFI$="x"THEN 6965
6955 PRINTI$;:INPUTVC:X(2)=X(2)+VC
6960 GOTO 6945
6965 CLS:INPUT"Sales price per unit";X(3
      )
6970 X(4)=INT(X(1)/(X(3)-X(2)))
6975 IF X(4)<0 THEN X$(4)="Error"ELSEX$(
      4)=STR$(X(4))
6980 CLS
6985 IF Z=0 THENPRINT"0 - Calculate prof
      it"ELSEPRINT"0 - End"
6990 PRINT"1 - Fixed costs      $";X(1)
6995 PRINT"2 - Variable costs  $";X(2)
7000 PRINT"3 - Sales price      $";X(3)
7005 IFZ=0 THEN PRINT"      Breakeven Unit
      s ";X$(4) ELSEPRINT"4 - Units sold
      ";X(4):PRINT"      Profit (-=loss) $";X(
      5)
7010 PRINT@240,"Change # or 0";:INPUTX
7015 IF(X>3 AND Z=0) OR (X>4 AND Z=1) TH
      EN 7010
7020 IFX=0 THEN IF Z=0 THEN 7045ELSEEND
7025 PRINT@X*40,CHR$(154);
7030 PRINT@240,"Enter new value ";:INPUT
      Y
```



```
7035 X(X)=Y
7040 IF Z=0 THEN GOT06970 ELSE 7045
7045 Z=1: X(5)=(X(3)*X(4))-(X(1)+(X(2)*X(
4))): GOT06980
```


INVOICE

This is an extremely useful program for creating and logging invoices. After the appropriate information is entered for each sale, you are given the option of printing an invoice in triplicate on an 80-column printer or saving the invoice in a .DO file to print out later.

Invoices are sequentially numbered and saved in a master invoice file. The program creates and uses two files:

1. INV.DO. This file stores your company name and summaries of all invoices printed in the form: invoice number/company name/date/amount.
2. IV#.DO. This is the text of the invoice. Invoice 10 will be saved as IV10.DO.

First, the program checks for the INV.DO file with the name of your company and the last invoice number used. If the file is not present, the program asks you for this information (sample answers are shown in bold type):

- 1—Your Co. Name? **ABC Corp.**
- 2—Your address? **111 Main Street**
- 3—Your C, S, Zip? **Manchester, NH 03103**

This last information may be entered with or without commas. This information will be printed on the top of the invoice. The computer will display:

Last invoice # was 0
This invoice will be 1
Press <ENTER> if correct or enter new Invoice
#. ?

If you press <ENTER>, this invoice will be numbered 1, or you may enter any number you wish. If you enter a different number, then the next invoice will be that number plus 1.

Now the program prompts for the customer information (again, sample answers are shown in bold type):

- 4—Sold to **Jane Jones**
- 5—Company? **Jones Co.**

- 6—Address? 123 West Street
7—City, State, Zip Code
? New York, NY 10021
8—Ordered by or purchase order number
? verbal order per "Jane" 12/12/83

Any information may be entered, up to 60 characters in length.

- 9—Terms or Paid . . .
? Paid VISA #5555 555 5555 555 1/08

This question may be answered with terms such as *net 10 days*, *2% cash*, *on account*, or other data. Commas are acceptable. If, though, the first four letters of the answer are *Paid*, *paid*, or *PAID*, as in the above example, then the next question (*Amount paid?*) will be asked. If any answer other than *paid* is given, then this question will be skipped.

- 10—Amount paid? 55.00

This question will be asked only if the first four letters of the answer to question 9 were *paid*. The amount paid will later be subtracted from the total amount due.

- 11—Other credits or (-) past due.
<ENTER> if none. ?

If the customer has a discount coupon or credit on his or her account that you would like to apply to this invoice, enter the amount here. If the customer has a past due balance and you would like it to appear on this invoice, enter it as a negative number. *Warning:* The past due amount will be added to the total amount of the invoice and will be stored in the INV.DO file. If a previous invoice was issued, then the past due amount will appear twice in your records: first in the original invoice and again in this invoice. This could cause your total sales, as stored in the INV.DO file, to be overstated.

- 12—Explanation?

If any value was given for the previous question, this question will be asked. Otherwise, it will be skipped. The answer to this question will be printed next to the previous answer. A typical answer

might be *50.00 discount coupon, return credit #1234, or past due amount.*

13—Message for bottom of invoice
?

You can have any short message printed at the bottom of the first copy of the invoice. If you just press <ENTER>, then the message *Thank you for your order* will be printed.

14—Date sold? 12/12/83

Enter this in any form you choose. *December 12, 1983* is also acceptable.

15—Salesman? **Mark Robinson**

16—Discount? 10

Many businesses give certain customers a discount on orders. Wholesalers typically list retail prices on invoices and give their distributors 45 or 50 percent off retail for their profit. If an amount is entered here, all prices will be calculated accordingly, and a line will appear on the invoice:

DISCOUNT 10% off retail

If just <ENTER> is pressed, then that line will not appear, and all prices will be unchanged. If you apply a different discount to every product, then just press <ENTER> and figure the appropriate discount into the price later.

The computer will display:

Press <ENTER> or enter a zero (0) for
Quantity when done.

17—Quantity? 2

Enter the quantity of the first item sold. Any number between 1 and 9999 is acceptable. If your quantity is 10,000 or more, then enter 10, begin your product description (the next question) with *per thousand*, and enter the price per thousand when that information is requested.

18—Product?

Enter the product name. Commas are acceptable.

19—Retail?

Enter the retail price per unit of the item, even if this is not the price the customer will finally be charged. If you are selling to a dealer or store, then the customer can use this column to determine what price he or she is going to use to sell the product.

20—Price or <ENTER> if calculated
?

Enter the per-unit price that the customer is to pay for the item. If the customer will be paying the retail price less the discount entered previously, simply press <ENTER>. For example, if the discount was 10 percent, the quantity was 2, and the retail price was \$100, then pressing <ENTER> will cause the customer to be charged \$180 (100 less 10 percent, times 2).

If there was no discount entered and you just press <ENTER>, then the customer will be charged full retail.

This question is most useful when the customer receives different discounts on different products. If, for example, product 1 retails for \$100 and the customer pays \$50, while product 2 retails for \$150 and the customer pays \$90, then the individual costs to the customer can be entered here.

The computer will then keep asking for *Quantity* until 0 or <ENTER> is answered.

Note: Up to eleven items may be entered on an invoice. If you will require more, change the value of the variable W in line 6300 to reflect the higher value. As the invoices get bigger, you may need more room to store all the data. If you get an *OS Error* (out of string space), increase the value of the CLEAR statement in line 6300.

21—Would you like a printout (Y/N)

If you answer Y, then an invoice, in triplicate, will be printed on your 80-column printer. Copy 2 will say at the bottom: PLEASE RETURN THIS COPY WITH YOUR PAYMENT. Copy 3 will say: FILE COPY.

22—Would you like the data saved to
a .DO file (Y/N)

If you answer Y, a .DO file will be created for printing on an 80-column printer later.

In every case, the INV.DO file will be appended to store the invoice summary. The first character in the INV.DO file will always be an asterisk (*). This is a pointer used by the program.

Line Number and Variable Cross-reference

Ref	Line Numbers				
+-----+					
:06330:	6330				
:06335:	6345				
:06350:	6340				
:06435:	6440				
:06445:	6445				
:06480:	6565				
:06485:	6485				
:06495:	6495				
:06540:	6530				
:06555:	6545				
:06570:	6825	6875			
:06660:	6650				
:06770:	6485				
:06810:	6810				
:06835:	6815				
:06840:	6840				
:06880:	6845				
:06885:	6305				
:A\$:	6700	6705	6710	6715	6720
:	6725				
:	PRINTUSING format field				
:C\$(:	6395	6400	6405	6410	6605
:	6610	6615	6620	6775	
:	Customer information				
:CC\$:	6775	6780			
:	Used if no company name				
:CR :	6425	6710	6725	6780	
:	Amount paid				
:D :	6460	6515	6650		
:	Discount				
:D\$:	6450	6630	6780		
:	Date				
:DI\$:	6460	6655			
:	Discount				
:G :	6320	6770			
:	Previous use pointer				
:G\$(:	6315	6320	6330	6580	6585
:	6590	6755	6770		
:	Seller information				
:GT :	6545	6705	6725	6780	


```
:      Grand total
:I$   : 6810   6815   6840   6845
:      Keyboard input
:LI$( : 6305   6560   6685
:      Line to print
:LP   : 6570   6650   6655   6660   6665
:      6675   6685   6695   6705   6710
:      6715   6720   6725   6730   6735
:      6740
:      Lineprinter position
:ME$  : 6445   6735
:      Bottom of invoice message
:NO   : 6385   6570   6780   6850
:      Invoice number
:NO$  : 6850   6855
:      Invoice number
:O$   : 6415   6640
:      Order number, or P.O. number
:OX$  : 6780   6790
:      Output file information
:PC#( : 6305   6510   6515   6520   6525
:      6530   6535   6545
:      Price
:PC$( : 6305   6530   6535   6540   6560
:      Price
:PR$( : 6305   6495   6560
:      Product descriptions
:QL   : 6750   6755   6825
:      Page number
:QN   : 6355   6370   6375   6385
:      Invoice Number
:QR   : 6430   6435   6715   6720   6725
:      6780
:      Other credits to account
:QR$  : 6435   6440   6715   6720
:      Other credits to account
:QR$( : 6715   6720
:      Other credits to account
:QU$( : 6305   6490   6560
:      Quantity
:QU(  : 6305   6485   6490   6545
:      Quantity
:RE$( : 6305   6500   6505   6560
:      Retail price
:RE(  : 6305   6500   6515
```

```

:      Retail price
:SS   : 6455  6645
:      Salesman
:TS   : 6420  6425  6635
:      Terms
:TI   : 6670  6680
:      Total items
:TQ$( : 6305  6545  6550  6555  6560
:      Total price
:TQ(  : 6305  6545  6550
:      Total price
:TT$  : 6465  6665
:      Header
:W    : 6300  6305
:      Number of items possible
:X    : 6325  6330  6350  6355  6470
:      6480  6485  6490  6495  6500
:      6505  6510  6515  6520  6525
:      6530  6535  6540  6545  6550
:      6555  6560  6670  6680  6685
:      Loop counter
:X$   : 6335  6350  6355
:      Information from INV.DO
:XX   : 6575  6595  6660  6695  6730
:      6740
:      Output file
:Y    : 6330
:      Used to format data
:YP   : 6325  6330
:      Used to format data
:ZZ   : 6495
:      Miscellaneous variable
+-----+

```


Invoice Program

```

6300 CLEAR2560:MAXFILES=2:W=11:REM MAXIM
UM NUMBER OF ITEMS
6305 ON ERROR GOTO 6885:DIM LI$(W):PC$(W
),PC$(W),PR$(W),QU(W),QU$(W),RE$(W),RE(W
),TQ(W),TQ$(W)
6310 OPEN"RAM:INV.DO"FORINPUTAS1
6315 INPUT#1,G$(1)
6320 IFG$(1)<>"*"THENG=1:LINEINPUT"Your
Co. Name? ";G$(1):LINEINPUT"Your address
? ";G$(2):LINEINPUT"Your C,S,Zip? ";G$(3
) ELSEINPUT#1,G$(1):INPUT#1,G$(2):INPUT#
1,G$(3)
6325 FORX=1TO3:YP=1
6330 Y=INSTR(YP,G$(X),","):IFY>0THENMIDS
(G$(X),Y,1)=" ":YP=Y+1:GOTO6330 ELSENEXT
6335 INPUT#1,X$
6340 IFEOF(1)THEN6350
6345 GOTO6335
6350 X=INSTR(1,X$,"/")
6355 QN=VAL(LEFT$(X$,X-1))
6360 CLOSE
6365 ON ERROR GOTO 0
6370 PRINT"Last Invoice # was ";QN
6375 PRINT"This Invoice will be";QN+1
6380 PRINT"Press <ENTER> if correct or e
nter new Invoice #.";
6385 INPUT NO:IF NO=0 THEN NO=QN+1
6390 CLS
6395 LINEINPUT"Sold to?";C$(1)
6400 LINEINPUT"Company?";C$(2)
6405 LINEINPUT"Address?";C$(3)
6410 LINEINPUT"City, State, Zip code
?";C$(4)
6415 LINEINPUT"Ordered by or purchase or
der number ?";O$
6420 LINEINPUT"Terms or Paid...
?";T$
6425 IFLEFT$(T$,4)="PAID" OR LEFT$(T$,4)
="paid"OR LEFT$(T$,4)="Paid"THEN INPUT"A
mount paid ";CR
6430 INPUT"Other credits or (-)past due.
<ENTER> if none. ";QR
6435 IF QR<>0 THEN LINEINPUT"Explainatio

```



```

n? ";QR$
6440 IF LEN(QR$)>60 THEN PRINT"60 chr ma
x.":GOTO 6435
6445 LINEINPUT"Message for bottom of inv
oice ";ME$:IFLEN(ME$)>60 THEN
PRINT"60 LETTERS MAXIMUM, PLEASE.":GOTO6
445
6450 CLS:INPUT"Date sold";D$
6455 INPUT"Salesman ";S$
6460 INPUT"Discount ";D:DI$=STR$(D)+"% O
FF RETAIL"
6465 TT$="QUAN DESCRIPTION
                                RETAIL    PRICE
TOTAL"
6470 X=0
6475 CLS:PRINT"Press <ENTER> or enter a
zero (0) for Quantity when done."
6480 X=X+1
6485 INPUT"Quantity";QU(X):IFQU(X)=0 THE
N 6770 ELSE IF QU(X)>9999THEN 6485
6490 QU$(X)=STR$(QU(X)):QU$(X)=QU$(X)+ST
RING$((6-LEN(QU$(X))), " ")+ " "
6495 PRINT"Product? ";STRING$(29, ".");:Z
Z=CSRLIN:PRINT@(((ZZ+1)*40)-31),"";:LINE
INPUT PR$(X):IFLEN(PR$(X))>29 THEN 6495
ELSE:PR$(X)=PR$(X)+STRING$((43-LEN(PR$(X
))), " ")+ " "
6500 INPUT"Retail";RE(X):RE$(X)=STR$(RE(
X)):IFRE(X)=INT(RE(X))THENRE$(X)=RE$(X)+
".00"
6505 RE$(X)=RE$(X)+STRING$((8-LEN(RE$(X)
)), " ")+ " "
6510 INPUT"Price or <ENTER> if calculate
d ";PC$(X)
6515 IFPC$(X)=0 THEN PC$(X)=INT(RE(X)*(1
00-D))/100
6520 PC$(X)=PC$(X)*100:PC$(X)=INT(PC$(X)
)
6525 PC$(X)=PC$(X)/100
6530 PC$(X)=STR$(PC$(X)):IFPC$(X)=INT(PC
$(X))THENPC$(X)=PC$(X)+".00":GOTO6540
6535 IF PC$(X)*10=INT(PC$(X)*10)THENPC$(
X)=PC$(X)+".0"
6540 PC$(X)=PC$(X)+STRING$((8-LEN(PC$(X)
)), " ")+ " "

```



```

6545 TQ(X)=PC$(X)*QU(X):GT=GT+TQ(X):TQ$(
X)=STR$(TQ(X)):IF TQ(X)=INT(TQ(X))THENTQ
$(X)=TQ$(X)+".00":GOTO6555
6550 IFTQ(X)*10=INT(TQ(X)*10)THEN TQ$(X)
=TQ$(X)+"0"
6555 TQ$(X)=TQ$(X)+STRING$((9-LEN(TQ$(X):
)), " ")+ " "
6560 LI$(X)=QU$(X)+PR$(X)+RE$(X)+PC$(X)+
TQ$(X)
6565 GOTO 6480
6570 LP=0:PRINT#1,"INVOICE";TAB(60);"NUM
BER ";NO
6575 FORXX=1TO4:PRINT#1,"":NEXT
6580 PRINT#1,"From:
      ";G$(1)
6585 PRINT#1,"
      ";G$(2)"
6590 PRINT#1,"
      ";G$(3)
6595 FORXX=1TO4:PRINT#1,"":NEXT
6600 PRINT#1,"Sold to:"
6605 PRINT#1,"          ";C$(1)
6610 PRINT#1,"          ";C$(2)
6615 PRINT#1,"          ";C$(3)
6620 PRINT#1,"          ";C$(4)
6625 PRINT#1,""
6630 PRINT#1,"Date sold:          ";D$
6635 PRINT#1,"Terms          ";T$
6640 PRINT#1,"Ordered by or PO#  ";O$
6645 PRINT#1,"Sold by          ";S$
6650 LP=23:IF D=0 THEN 6660
6655 PRINT#1,"Dealer Discount    ";DI$:L
P=LP+1
6660 FORXX=1TO4:PRINT#1,:LP=LP+1:NEXT
6665 PRINT#1, STRING$(79,"="):PRINT#1,""
:PRINT#1, TT$:PRINT#1,STRING$(79,"-"):LP
=LP+4
6670 TI=X
6675 PRINT#1,"":LP=LP+1
6680 FORX=1TO TI
6685 PRINT#1, LI$(X):LP=LP+1
6690 NEXT
6695 FORXX=1TO4:PRINT#1,"":LP=LP+1:NEXT
6700 A$="$###,###.##"
6705 PRINT#1,"
      Total Order    ";:PRINT#1, U

```



```
SING A$;GT:LP=LP+1
6710 IF CR>0 THEN PRINT#1,"
                        Paid with order  -
";:PRINT#1,USINGA$;CR:LP=LP+1
6715 IF QR>0 THEN QR$(1)=STRING$((50-LEN
(QR$)), " ") + QR$ + " - ";:PRINT#1,QR$(1) ;:P
RINT#1,USINGA$;QR:LP=LP+1
6720 IF QR<0 THEN QR$(1)=STRING$((50-LEN
(QR$)), " ") + QR$ + " ";:PRINT#1,QR$(1) ;:P
RINT#1,USINGA$;ABS(QR):LP=LP+1
6725 PRINT#1,"
                        TOTAL DUE      ";:PRINT#1,US
INGA$;GT-CR-QR:LP=LP+1
6730 FORXX=1 TO 4:PRINT#1,"":LP=LP+1:NEX
T
6735 IFME$="" THEN PRINT#1,"Thank you for
your order":LP=LP+1:ELSE PRINT#1,ME$:LP
=LP+1
6740 FORXX=1 TO 63-LP
6745 PRINT#1,"":NEXT
6750 IF QL=2 THEN PRINT#1,"PLEASE RETURN
THIS COPY WITH YOUR PAYMENT"
6755 IF QL=3 THEN PRINT#1, G$(1);" FILE
COPY"
6760 PRINT#1,"":PRINT#1,""
6765 RETURN
6770 IFG=1 THEN OPEN"INV.DO"FOR OUTPUTAS1:
PRINT#1,"*":PRINT#1,G$(1):PRINT#1,G$(2):
PRINT#1,G$(3):CLOSE
6775 IFC$(2)="" THEN C$=C$(1) ELSE C$=C$(2
)
6780 OX$=STR$(NO)+"/"+CC$+ "/" + D$ + "/" + STR
$(GT-CR-QR)
6785 OPEN"INV.DO"FOR APPENDAS1
6790 PRINT#1,OX$;
6795 CLOSE
6800 CLS
6805 PRINT"Would you like a printout (Y/
N)"
6810 I$=INKEY$:IFI$="" THEN 6810
6815 IFI$="n" OR I$="N" THEN 6835
6820 OPEN"LPT:"FOR OUTPUTAS1
6825 FORQL=1 TO 3:GOSUB 6570:NEXT
6830 CLOSE
6835 PRINT"Would you like data saved to
a .DO file (Y/N)"
```



```
6840 I$=INKEY$:IFI$=""THEN6840
6845 IFI$="n"ORI$="N"THEN6880
6850 NO$=STR$(NO):NO$=RIGHT$(NO$,LEN(NO$
)-1):IFLEN(NO$)>4THENNO$=RIGHT$(NO$,4)
6855 FN$="IV"+NO$
6860 FN$="RAM:"+FN$+".DO"
6865 PRINT"Data saved in ";FN$
6870 OPEN FN$ FOROUTPUTAS1
6875 GOSUB6570
6880 CLOSE:MAXFILES=0:CLEARO:MENU
6885 IFERR=52THENOPEN"INV.DO"FOROUTPUTAS
1ELSEONERRORGOTO0:RESUME
6890 PRINT#1,"/":PRINT#1,"O/":CLOSE:RESU
ME
```

JOB QUOTE

The Job Quote program is a selling aid. It allows a representative to specify and price a job on the spot. If a printer is available, the sales rep can offer the potential customer a written quotation. Without a printer, the program stores the relevant information in a .DO file for later review.

In the past, I used the program to develop a price on security and fire alarm systems for my customers. I kept my Model 100 in a briefcase along with the small Radio Shack CGP-115 printer and cable. All I needed to do was plug in the printer.

The program will work with any type of business that combines equipment and labor to make a final product or service.

When you execute the program, it first looks for a file in memory called NAME.DO containing information on your own business. If the file is not present, the initial questions will be asked and the NAME.DO file created.

These questions are (with sample answers shown in bold type):

- 1—Your company name? **ABC Corp.**
- 2—Your company address? **111 Main Street**

Do not use commas in the answer.

- 3—Your city, state, zip? **Manchester, NH, 03103**

Enter the three items, separating them with commas.

- 4—Your phone number? **(603) 555-1212**

Type your phone number as you would like it to appear on the quotation. Any format that does not include commas is acceptable.

- 5—Hourly labor rate? **25**

This information will now be saved in a file called NAME.DO. The NAME.DO file will look like this if you examine it with TEXT:

ABC Corp.
111 Main Street
Manchester

NH
03103
(603) 555-1212
25

If this company information is already on file in the computer's RAM, then the starting question will be (again, sample answers are shown):

6—Customer name? **XYZ Corp.**
7—Company? **111 Main Street Corp.**
8—Address? **111 Main Street**

Type the information using no commas.

9—City, State, Zip? **New York, NY, 10021**

Enter the three items, separated by commas.

10—Phone? **(212) 555-1212**
11—Other? **Attn: Jane Jones**

The *Other* question can be used for any important piece of customer information desired. Do not use commas.

12—Type of job? **Burglar Alarm System**

Any appropriate information may be entered, without commas, please.

13—Quan., item? **2, Motion detectors**

Begin entering your list of materials.

14—Cost? **245.00**

Enter the *per-item* cost of the item previously entered. In this example, motion detectors cost \$245 each.

15—Labor hours? **2**

Enter the labor required to install one motion detector. When calculated later, the labor hours will be multiplied by the quantity

entered above and the hourly labor rate entered during the initial question section.

The computer will respond with:

2—Motion detectors 490 4

This displays the total cost and labor hours.

The program will continue with the *Quan., item?* prompt until you press <ENTER> in answer to it without entering any data.

16—Misc. costs? 100.00

Enter an amount, if needed, for difficulty, travel, or other charges to the customer. Press <ENTER> if none.

17—Misc. labor hrs.? 12

Enter any additional labor needed. Press <ENTER> if none.

The computer will then print the totals on the screen:

Equipment = \$590.00

Labor = \$400.00

TOTAL COST = \$990.00

18—Quote? 950

Enter the price you wish to give to the customer. In this case the sales rep decided to give the customer a few dollars off the regular price.

19—Lineprinter (Y/N) y

If you have a printer connected and ready and this question is answered *y* or *Y*, then a printout like the following will be made. This program will not use more than 40 characters per line.

ABC Corp.
111 Main Street
Manchester, NH, 03103
(603) 555-1212

.....

XYZ Corp.
111 Main Street Corp.
111 Main Street
New York, NY, 10021
(212) 555-1212
Attn: Jane Jones
Burglar Alarm System
Work to consist of:

2—Motion detectors
.
.
.
[remaining items]
TOTAL COST \$950.00

This quotation can be left with the customer. The charges for miscellaneous costs or labor, and the total calculated price, are not shown, only the equipment list and total quote.

The program creates a file in which the quotation is stored with itemized costing information. The name of this new file is the first 6 characters of the company name. In this example, it would be 111 MA.DO. The file will look like this:

XYZ Corp.
111 Main Street Corp.
111 Main Street
New York, NY, 10021
(212) 555-1212
Attn: Jane Jones
Burglar Alarm System
2—Motion detectors
Misc. cost \$100
Misc. labor \$300
Total cost \$590
Total labor \$400
Price \$990
Quote \$950

Job Quote is designed to operate with eleven items total. If you wish more, change the value of the variable W in line 5855. As Job Quote gets bigger, you may need more room to store the data. If you get an *OS Error* (out of string space), increase the value of the CLEAR statement in line 5855.

Line Number and Variable Cross-reference

Ref	Line Numbers				
+-----+					
:05910:	5950				
:05955:	5915				
:06030:	6015				
:06060:	6050				
:06075:	6065				
:06090:	6080				
:06135:	5860				
:06160:	6210				
:06165:	6135				
:A\$(5865	5870	5875	5880	5885
:	5890	6000	6030	6035	6040
:	6045				
:	Customer information				
:C(5855	5920	5930	5940	5945
:	6090				
:	Cost				
:C1	5955	5960	6105		
:	Miscellaneous costs				
:E\$(5855	5910	5915	5935	5940
:	6085	6090			
:	Equipment				
:L(5855	5925	5930	5940	5945
:	6090				
:	Labor hours				
:L1	5965	5970	6110		
:	Miscellaneous labor				
:LB	5980	5990	6110	6150	6190
:	6205				
:	labor per item				
:LP	6010	6015	6030	6040	6050
:	6065	6080	6100		
:	Lineprinter flag				
:N	5910	5930	5935		
:	Quantity				
:NA\$(6020	6145	6170	6175	6180
:	6185	6200			
:	Seller information				
:Q	5995	6100	6125		
:	Quote				
:S\$(5895	6055	6060		


```
:      Type of job
:TC   : 5905   5945   5960   5975   5990
:      6115   6120
:      Total cost
:TL   : 5905   5945   5970   5980   5990
:      6120
:      Total labor
:W    : 5855
:      Number of items permitted
:X    : 5905   5910   5915   5920   5925
:      5930   5935   5940   5945   6075
:      Loop counter
:X$   : 6010
:      Keyboard input
:Z    : 6030   6035   6075   6085   6090
:      6145   6200
:      Loop counter
+-----+
```


Job Quote Program

```

5850 MAXFILES=2
5855 CLEAR 2000:W=11:DIM E$(W):L(W),C(W)
5860 GOSUB 6135
5865 INPUT"Customer name";A$(1)
5870 INPUT"Company";A$(2)
5875 INPUT"Address";A$(3)
5880 INPUT"City, State, Zip";A$(4),A$(5)
,A$(6)
5885 INPUT"Phone";A$(7)
5890 INPUT"Other";A$(8)
5895 INPUT"Type of job";S$(1)
5900 PRINT"Work to consist of:"
5905 X=1:TC=0:TL=0
5910 INPUT"Quan, item";N,E$(X)
5915 IF E$(X)=""THENX=X-1:GOTO5955
5920 INPUT"Cost";C(X)
5925 INPUT"Labor hours";L(X)
5930 IFN>0THENC(X)=C(X)*N:L(X)=L(X)*N
5935 E$(X)=STR$(N)+"-"+E$(X)
5940 PRINTE$(X);C(X);L(X)
5945 TC=TC+C(X):TL=TL+L(X):X=X+1
5950 GOTO5910
5955 INPUT"Misc. costs";C1
5960 TC=TC+C1
5965 INPUT"Misc. labor hrs.";L1
5970 TL=TL+L1
5975 PRINT"Equipment = $";TC
5980 PRINT"Labor      = $";TL*LB
5985 PRINT"          -----"
5990 PRINT"TOTAL COST= $";TC+(TL*LB)
5995 PRINT:INPUT"Quote";Q
6000 FN$="RAM:"+LEFT$(A$(2),6)+".DO"
6005 OPEN FN$ FOR OUTPUT AS 1
6010 INPUT"Lineprinter (Y/N)";X$:IF X$="
Y" OR X$="y" THEN LP=1
6015 IFLP=0THEN 6030
6020 LPRINT NA$(1):LPRINT NA$(2):LPRINT
NA$(3);", ";NA$(4);", ";NA$(5):LPRINT NA
$(6)
6025 LPRINT"....."
6030 FORZ=1TO3:IFLP=1THENLPRINTA$(Z)
6035 PRINT#1,A$(Z):NEXT .
6040 IFLP=1THENLPRINTA$(4);", ";A$(5);",
";A$(6):LPRINTA$(7):LPRINTA$(8)

```



```
6045 PRINT#1,A$(4);", ";A$(5);", ";A$(6)
:PRINT#1,A$(7):PRINT#1,A$(8)
6050 IFLP=1THENLPRINT:LPRINTELSE6060
6055 LPRINTS$(1)
6060 PRINT#1,S$(1)
6065 IFLP=1THENLPRINT"Work to consist of
:"ELSE6075
6070 LPRINTSTRING$(40,"-")
6075 FORZ=1TOX
6080 IFLP=1THENLPRINTELSE6090
6085 LPRINT$(Z);
6090 PRINT#1,E$(Z);C(Z);L(Z)
6095 NEXT
6100 IFLP=1THENLPRINT:LPRINT"TOTAL COST
";:LPRINTUSING"$$,#####.##";Q
6105 PRINT#1,"Misc Cost $";C1
6110 PRINT#1,"Misc Labor$";L1*LB
6115 PRINT#1,"Total Cost$";TC
6120 PRINT#1,"Total Labor";TL:PRINT#1,"P
rice ";TL+TC
6125 PRINT#1,"Quote";Q
6130 END
6135 ON ERROR GOTO 6165
6140 OPEN"RAM:NAME.DO" FOR INPUT AS 2
6145 FOR Z=1 TO 6:INPUT#2,NA$(Z):NEXT
6150 INPUT#2,LB
6155 CLOSE
6160 RETURN
6165 IF ERR<>52 THEN ON ERROR GOTO 0:RES
UME
6170 INPUT"Your Company name";NA$(1)
6175 INPUT"Your company address";NA$(2)
6180 INPUT"Your city, state, zip
";NA$(3),NA$(4),NA$(5)
6185 INPUT"Your phone number";NA$(6)
6190 INPUT"Hourly labor rate $";LB
6195 OPEN"RAM:NAME.DO" FOR OUTPUT AS 2
6200 FOR Z=1TO6:PRINT#2,NA$(Z):NEXT
6205 PRINT#2,LB
6210 RESUME 6160
```


CALENDAR

Calendar is a simple, yet useful, utility for printing a calendar for any month, any year. Simply enter the month and year for which you want a calendar, and it will appear on the display. If you press <ENTER> instead of entering a date, you'll get a calendar for the month and year as indicated on the Model 100 menu (make sure that the current date in the Model 100 clock is correct, or your calendar will be for a month other than the correct one).

If you have a printer connected and ready, press the PRINT function key to receive a printout of the calendar display.

If you want to use Calendar as a subroutine in another program, all you have to do is merge it with the other program and add two lines:

```
1 KEY(1)ON:ON KEY GOSUB8620
8620 DA%=0:D$=" "
```

Make sure that none of the line numbers of the Calendar routine write over any of the lines of your program. Line 1 must be included for this routine to work inside another program. Make sure none of the variables used in Calendar affect the variables in your program.

When you run your program with the Calendar routine in it and you want to see the calendar, press the F1 key. The calendar will appear on your display. Press <ENTER> to return to your program.

Line Number and Variable Cross-reference

Ref	Line Numbers					
+-----+						
:08620:	8630	8640	8740			
:08725:	8725					
:08735:	8610					
:08745:	8655					
:C\$: 8725					
:	Keyboard input					
:D	: 8685	8760				
:	Day					
:D\$: 8620	8625	8630	8635		
:	Date					
:D%	: 8685	8690	8715			
:	Day					
:DA%	: 8620	8625	8700	8710		
:	Day					
:I%	: 8630	8635	8660	8695	8700	
:	8705	8710	8715	8720	8720	
:	Used in date decoding					
:M\$: 8660	8675				
:	Months					
:MD%	: 8660	8665	8695	8715		
:	Days in months					
:V	: 8635	8640	8645	8665	8675	
:	8750	8760				
:	Year					
:W	: 8635	8640	8660	8665	8745	
:	8755					
:	Month					
:X	: 8635	8750				
:	Calculation variable					
:Y	: 8745	8750	8755	8760		
:	Calculation variable					
:Z	: 8750	8755	8760			
:	Calculation variable					
+-----+						

Calendar Program and Subroutine

```

8600 ' with permission from
      Michael M. Rubenstein
      Compuserve <70205,1144>
8605 CLEAR 1024
8610 ON ERROR GOTO 8735
8615 PRINT:PRINT"Press <ENTER> for curre
nt month or enter"
8620 DA%=0:D$="":INPUT "Month/year (MM/Y
Y)";D$
8625 IF D$="" THEN D$=DATE$:DA%=VAL(MID$
(D$,4,2)):D$=LEFT$(D$,3)+RIGHT$(D$,2)
8630 I%=INSTR(D$,"/"):IF I%<2 OR I%>3 OR
I%=LEN(D$) THEN BEEP:GOTO 8620
8635 W=VAL(LEFT$(D$,I%-1)):V=VAL(RIGHT$(
D$,LEN(D$)-I%)):X=1
8640 IF W<1 OR W>12 OR V<0 THEN BEEP:GOT
O 8620
8645 IF V<100 THEN V=V+1900
8650 ON ERROR GOTO 0
8655 GOSUB 8745
8660 FOR I%=1 TO W:READ M$,MD%:NEXT I%
8665 IF W=2 THEN MD%=MD%-(0=V MOD 4)+(0=
V MOD 100)-(0=V MOD 400)
8670 CLS
8675 PRINT TAB(7);M$;" ";V
8680 PRINT TAB(6);" Sun Mon Tue Wed Thu
Fri Sat"
8685 D%=D-7*INT(D/7)
8690 PRINT TAB(6);SPACE$(4*D%);
8695 FOR I%=1 TO MD%
8700 IF I%=DA% THEN PRINT CHR$(27);"p";
8705 PRINT USING "####";I%;
8710 IF I%=DA% THEN PRINT CHR$(27);"q";
8715 D%=D%+1:IF D%>6 AND I%<>MD% THEN PR
INT:PRINT TAB(6);:D%=0
8720 NEXT I%
8725 C$=INKEY$:IF C$="" THEN 8725
8730 MENU
8735 IF ERL<>60 THEN ON ERROR GOTO 0:RES
UME
8740 BEEP:RESUME 8620
8745 Y=-INT ((14-W)/12)
8750 Z=X-32074+INT (1461*(V+4800+Y)/4)
8755 Z=Z+INT (367*(W-2-12*Y)/12)

```



```
8760 D=Z-INT (3*INT ((V+4900+Y)/100)/4)
8765 RETURN
8770 DATA January,31
8775 DATA February,28
8780 DATA March,31
8785 DATA April,30
8790 DATA May,31
8795 DATA June,30
8800 DATA July,31
8805 DATA August,31
8810 DATA September,30
8815 DATA October,31
8820 DATA November,30
8825 DATA December,31
```

DAYS BETWEEN DATES

This program is actually a *subroutine* designed to be used, normally, by another program to count how many days there are between two given dates.

The main program, in such an instance, would call this program with a GOSUB 7065 command.

Before using this program, you must set up the following variables:

The future-most date:

M1 = Future-most month

D1 = Future-most day

Y1 = Future-most year

The lesser date:

MM = Lesser month

DD = Lesser day

YY = Lesser year

This subroutine is used in the Weighted Average Number of Stock Shares Outstanding program (chapter three). When this program returns control back to the main program, the variable DB will hold the number of days between the two dates.

To use this subroutine as an independent program, make these changes to the listing:

```
7050 INPUT "FUTURE-MOST MONTH, DAY,  
YEAR "; M1, D1, Y1  
7055 INPUT "ENTER LESSER MONTH, DAY,  
YEAR "; MM, DD, YY  
7175 PRINT "THERE ARE "; DB " DAYS BETWEEN  
"; M1 "/" ; D1 "/" ; Y1 " AND "; MM "/" ; DD "/" ; YY
```


Line Number and Variable Cross-reference

Ref	Line Numbers			
+-----+				
:07130:	7095	7145		
:07150:	7105	7130	7165	
:07170:	7085	7150		
:07175:	7170			
:D1	: 7065	7170		
:	Future-most day			
:DB	: 7090	7100	7115	7135 7155
:	7170			
:	Days between dates			
:DD	: 7065	7090	7095	7125 7170
:	Lesser days			
:M(: 7070	7080	7090	7115 7155
:	Days in month			
:M1	: 7065	7085	7150	
:	Future-most month			
:MM	: 7065	7085	7090	7095 7100
:	7110	7115	7125	7150 7155
:	7160			
:	Lesser month			
:X	: 7080	7110	7115	7120
:	Loop counter			
:Y1	: 7065	7085	7105	7130
:	Future-most year			
:YY	: 7065	7085	7090	7095 7100
:	7105	7115	7125	7130 7135
:	7140	7155		
:	Lesser year			
+-----+				

Days between Dates Program and Subroutine

```
7050 ' DAY BETWEEN DATES
      DBD
7055 ' ***** SUBROUTINE *****
7060 ' M1/D1/Y1 - MM/DD/YY
7065 IF YY>Y1 OR (YY=Y1 AND MM>M1) OR (Y
Y=Y1 AND MM=M1 AND DD>D1) THEN PRINT"DAT
E ENTRY ERROR":END
7070 DIM M(12)
7075 DATA 31,28,31,30,31,30,31,31,30,31,
30,31
7080 FOR X=1 TO 12:READM(X):NEXT X
7085 IF Y1=YY AND M1=MM THEN GOTO 7170
7090 DB=M(MM)-(DD-1):IF INT(YY/4)=YY/4 A
ND MM=2 THEN DB=DB+1
7095 DD=1:MM=MM+1:IFMM=13 THEN MM=1:YY=Y
Y+1:GOTO7130
7100 PRINTDB;" TO ";MM;"/01/";YY
7105 IF Y1=YY THEN 7150
7110 FOR X=MM TO 12
7115 DB=DB+M(X):IF INT(YY/4)=YY/4 AND MM
=2 THEN DB=DB+1
7120 NEXT X
7125 MM=1:DD=1:YY=YY+1
7130 IF Y1=YY THEN 7150
7135 DB=DB+365:IF INT(YY/4)=YY/4 THEN DB
=DB+1
7140 YY=YY+1
7145 GOTO 7130
7150 IFM1=MM THEN 7170
7155 DB=DB+M(MM):IF INT(YY/4)=YY/4 AND M
M=2 THEN DB=DB+1
7160 MM=MM+1
7165 GOTO7150
7170 IF DD=D1 THEN 7175 ELSE DB=DB+(D1-D
D)
7175 RETURN
```

2

Management Decision Making

Managers and other businesspeople use a number of methods to make decisions. Committee meetings, bar charts, computer analysis, consultations, pie charts, DECIDE analysis, flow charts, and decision trees are among the many forms and methods of correlating relevant information.

Often, the larger the organization, the more sophisticated the decision-making process. This does not necessarily imply accuracy or effectiveness. I can't help but wonder what decision-making process was used by the manufacturers of M&M candies when they refused to allow the use of their product in the movie *E.T.* (Reese's Pieces, the candy used in the movie, experienced an unprecedented jump in sales.)

The programs in this chapter can help you analyze the financial aspects of a decision. These programs allow you to compare different terms and rates; they simply provide information in a more understandable manner. In some cases, a program will make a recommendation. Note that there are many additional factors that influence a decision and that any such recommendation is based on very basic information and calculations.

TERM VERSUS WHOLE LIFE INSURANCE

When buying life insurance, you usually have two choices: *whole life* and *term*. *Whole life* insurance policies have a built-in savings or investment program and a specified cash value at the end of a specified period of time. *Term* policies don't build up to any cash value, but they do offer protection at a much lower premium rate.

Many advisors feel that you should buy term instead of whole life and put the difference in premiums into a savings account. In actuality, this depends on a number of factors, primarily whether you can invest your money at a better rate than the whole life insurance company offers.

Which you choose depends on your specific financial situation and goals, but this program will tell you which type of policy will give you the best results for a specified period of time.

The computer takes the premiums of the two policies and the whole life policy's ending cash value and compares the difference in premiums with putting money in a bank, with specific interest earning and compounding rates. Then it makes a suggestion as to which course of action you should take. The program assumes that the interest rate you supply remains constant over the entire period of time that the whole life policy is written to cover.

Examples

<i>Program</i>	<i>Your response</i>
Monthly premium of term policy?	4.6
Monthly premiums of whole life policy?	12
Enter the cash value of the whole life policy?	5000
After how many years?	25
Enter current annual interest rate?	9.97
Compounded how many times a year?	12
The option to your best advantage: Buy the term insurance and put the difference in the cost between the policies (\$7.4) into a savings account.	

You come out ahead by \$4,849.69.

Press <ENTER> to continue, M for Menu

Monthly premium of term policy?	9
Monthly premiums of whole life policy?	12
Enter the cash value of the whole life policy?	5000
After how many years?	25
Enter current annual interest rate?	6
Compounded how many time a year?	12
The option to your best advantage: Buy the whole life policy. Putting the difference in cost between the two pol- icies (\$3) in a savings account would net you \$2,910.62, less than the cash value of the policy.	

Press <ENTER> to continue, M for Menu

Line Number and Variable Cross-reference

Ref	Line Numbers
+-----+	
:03250:	3345
:03345:	3335
:09000:	3300
:10000:	3345
:C :	3295 9000 9005
:	Compounding periods
:CP :	9005 9015
:	Compounds per payment
:CV :	3270 3325
:	Cash value
:IC :	9000 9015
:	Interest rate/compounding period
:IP :	3310 3320 9020
:	Interest calculator
:IR :	3290 9000
:	Current interest rate
:N2 :	3285 9005
:	Compound periods per year
:N3 :	3285 3305
:	Months until maturity
:P# :	3310 3320 3325 3335 3340
:	Miscellaneous variable
:PQ :	3280 3310
:	Cost difference
:QV :	9015 9020
:	Effective interest rate
:T :	3275 3285
:	Years
:TL :	3260 3280 3335 3340
:	Term policy cost
:US :	3255 3335 3340
:	PINTUSING format field
:WL :	3265 3280 3335 3340
:	Whole life policy cost
:X :	3305 3315 9015
:	Loop counter
+-----+	

Term versus Whole Life Insurance Program

```

3250 CLS:PRINT"  COST OF TERM VS WHOLE L
IFE INSURANCE":REM D2
3255 U$="$ $ #,####.##"
3260 INPUT"Monthly premium of term polic
y";TL
3265 PRINT"Monthly premiums of whole lif
e":INPUT"policy";WL
3270 INPUT"Enter the cash value of the w
hole life  policy";CV
3275 INPUT"After how many years";T
3280 PQ=WL-TL
3285 N3=T*12:N2=12
3290 INPUT"Enter current annual interest
rate";IR
3295 INPUT"Compounded how many times a y
ear";C
3300 GOSUB9000:' FIND EFFECTIVE INTEREST
RATE PER PAYMENT
3305 FOR X=1 TO N3
3310 P#=PQ+(P#+(P#*((IP)/100))):' MOVE F
WD 1 PMT PERIOD
3315 NEXT X
3320 P#=P#+(P#*((IP)/100)):' MOVE FWD LA
ST PMT PERIOD
3325 P#=P#-CV
3330 CLS:PRINT"The option to your best a
dvantage:"
3250 CLS:PRINT"  COST OF TERM VS WHOLE L
IFE INSURANCE":REM D2
3255 U$="$ $ #,####.##"
3260 INPUT"Monthly premium of term polic
y";TL
3265 PRINT"Monthly premiums of whole lif
e":INPUT"policy";WL
3270 INPUT"Enter the cash value of the w
hole life  policy";CV
3275 INPUT"After how many years";T
3280 PQ=WL-TL
3285 N3=T*12:N2=12
3290 INPUT"Enter current annual interest
rate";IR

```



```
3295 INPUT"Compounded how many times a y
ear";C
3300 GOSUB9000:' FIND EFFECTIVE INTEREST
RATE PER PAYMENT
3305 FOR X=1 TO N3
3310 P#=PQ+(P#+(P#*((IP)/100))):' MOVE F
WD 1 PMT PERIOD
3315 NEXT X
3320 P#=P#+(P#*((IP)/100)):' MOVE FWD LA
ST PMT PERIOD
3325 P#=P#-CV
3330 CLS:PRINT"The option to your best a
dvantage:"
3335 IF P#>0 THEN PRINT"Buy the Term Ins
urance and put the dif- ference in the c
ostbetween the policies ($";WL-TL;" ) int
o a savings account.":PRINT"You come out
ahead by";:PRINTUSINGU$;P#;:PRINT".":GO
TO3345
3340 PRINT"Buy the Whole Life policy. P
utting the difference in cost between th
e two pol- icies ($";WL-TL;" ) in a savin
gs account":PRINT"would net you";:PRINTU
SINGU$;ABS(P#);",":PRINT"less than the c
ash value of the policy."
3345 GOSUB10000:RUN3250
9000 IC=(IR/C)/100:'INTEREST RATE PER CO
MPOUNDING PERIOD
9005 CP=C/N2:'COMPOUNDS PER PAYMENT
9010 'FIND EFFECTIVE INTEREST RATE FOR C
2 PERIODS
9015 QV=1:FOR X=1 TO CP:QV=QV+(QV*IC):NE
XT
9020 IP=(QV-1)*100:'IP=INTEREST PER PAYM
ENT
9025 RETURN
10000 PRINT@281,"Press <ENTER> to contin
ue, M for MENU";
10005 A$=INKEY$:IFA$="M"ORA$="m"THENMAXF
ILES=0:CLEARO:MENUELSEIFA$<>CHR$(13)THEN
10005ELSERETURN
```


TIME PAYMENT VERSUS PURCHASE

Occasionally when selling goods, a sales representative will offer you the opportunity to buy the items on a *time payment*, or *credit*, plan. Before buying anything, you should compare the payments and interest rate charged with the actual purchase price of the product. You might find that, while the plan may offer "easy credit," it isn't a bargain.

This program takes the cash price of the item, the amount and number of payments, the normal or average annual interest rate charged by banks and other lenders, the compounding period, and the number of payments per year and offers a suggestion as to whether you should buy the product on credit.

A completely neutral option would be a time payment total that exactly equals the cash price plus interest. In this case, it wouldn't matter financially which plan you used, commercial loan or time payment plan. In that situation, this program's answer would be \$0.00.

Examples

<i>Program</i>	<i>Your response</i>
Enter cash purchase price?	1000
Enter amount of payments?	25
Total number of payments?	100
Number of payments per year?	12
Enter annual interest rate %?	18
Compounded how many times a year?	12
Pay cash for the item; the interest rate is too high. You are being charged \$290.62 above the cost of the item plus interest.	

Press <ENTER> to continue, M for Menu

Enter cash purchase price?	6500
Enter amount of payments?	150
Total number of payments?	48
Number of payments per year?	12
Enter annual interest rate %?	5.25
Compounded how many times a year?	12
Accept the time payment offer. You would pay \$18.48 less than the cost of the item plus interest if you paid cash.	
Press <ENTER> to continue, M for Menu	

Line Number and Variable Cross-reference

Ref	Line Numbers
+-----+	
:03350:	3435
:03435:	3395 3425
:09000:	3400
:10000:	3435
:C :	3390 3395 9000 9005
:	Compounding frequency
:CA :	3365 3420
:	Purchase price
:CP :	9005 9015
:	Compounds per payment
:IC :	9000 9015
:	Interest rate/compound period
:IP :	3410 9020
:	Interest per payment
:IR :	3385 9000
:	Annual interest rate
:N2 :	3380 3395 9005
:	Number of payments per year
:N3 :	3375 3405
:	Total number of payments
:P# :	3410 3420 3425 3430
:	Present value of time payments
:PQ :	3370 3410
:	Amount of payments
:QV :	9015 9020
:	Effective interest rate
:U\$:	3355 3425 3430
:	PRINTUSING format field
:X :	3405 3415 9015
:	Loop counter
+-----+	

Time Payment versus Purchase Program

```
3350 CLS:PRINT"    COMPARING TIME PAYMENT
S TO PRICE":REM D3
3355 US="$ $#,#####.##"
3360 ' CHARGE IT AT MONTHLY PAYMENTS OF
# OR PURCHASE FOR ##
3365 INPUT"Enter cash purchase price";CA
3370 INPUT"Enter amount of payments";PQ
3375 INPUT"Total number of payments";N3
3380 INPUT"Number of payments per year";
N2
3385 INPUT"Enter annual interest rate %"
;IR
3390 INPUT"Compounded how many times a y
ear";C
3395 IF N2>C THENPRINT"YOU MAY NOT MAKE
MORE THAN ONE PAYMENT PER INTEREST COMP
OUNDING PERIOD. PLEASE RE-ENTER":GOTO343
5
3400 GOSUB9000:' FIND EFFECTIVE INTEREST
RATE PER PAYMENT
3405 FOR X=N3 TO 1 STEP -1
3410 P#=(PQ+P#)/(1+(IP/100)):'MOVE BACK
ONE PMT
3415 NEXT X
3420 P#=P#-CA
3425 IF P#>0 THEN PRINT"Pay cash for the
item, the interest rate is too high. Y
ou are being charged:":PRINTUSINGU$;P#;;
PRINT" above the cost of the":PRINT"item
plus interest.":GOTO3435
3430 PRINT"Accept the time payment offer
. You wouldpay ";:PRINTUSINGU$;ABS(P#);:
PRINT" less than the cost of the item pl
us interest if you pay cash."
3435 GOSUB10000:RUN3350
9000 IC=(IR/C)/100:'INTEREST RATE PER CO
MPOUNDING PERIOD
9005 CP=C/N2:'COMPOUNDS PER PAYMENT
9010 'FIND EFFECTIVE INTEREST RATE FOR C
2 PERIODS
9015 QV=1:FOR X=1 TO CP:QV=QV+(QV*IC):NE
XT
```



```
9020 IP=(QV-1)*100:'IP=INTEREST PER PAYM
ENT
9025 RETURN
10000 PRINT@281,"Press <ENTER> to contin
ue, M for MENU";
10005 A$=INKEY$:IFA$="M"ORA$="m"THENMAXF
ILES=0:CLEARO:MENUELSEIFA$<>CHR$(13)THEN
10005ELSERETURN
```


TOTAL INTEREST PAID ON A LOAN

When borrowing money, you need to determine exactly how much the loan is actually going to cost you. This program takes the amount of the loan, the number of payments per year, the total number of payments, the annual interest rate, and the compounding period and calculates how much money you are paying above the loan amount.

Examples

<i>Program</i>	<i>Your response</i>
Enter amount of loan?	10000
Total number of payments?	36
Number of payments per year?	12
Enter annual interest rate %?	12
Compounded how many times a year?	12
Total interest charge is \$1,956.74	
Press <ENTER> to continue, M for Menu	
Enter amount of loan?	7798
Total number of payments?	48
Number of payments per year?	12
Enter annual interest rate %?	9.8
Compounded how many times a year?	12
Total interest charge is \$1,677.26	
Press <ENTER> to continue, M for Menu	

Line Number and Variable Cross-reference

Ref	Line Numbers
+-----+	
:03450:	3545
:03495:	3515 3520
:03530:	3510 3520
:03545:	3525
:09000:	3485
:10000:	3545
:C :	3475 9000 9005
:	Compounding frequency
:CP :	9005 9015
:	Compounds per payment
:IC :	9000 9015
:	Interest rate/compound period
:IP :	3500 9020
:	Interest rate per payment
:IR :	3470 9000
:	Annual Interest
:N2 :	3465 9005
:	Number of payments per year
:N3 :	3460 3490 3495 3530
:	Total number of payments
:P# :	3500 3510 3515 3520
:	Miscellaneous variable
:PP :	3510
:	Miscellaneous variable
:PQ :	3490 3500 3515 3520 3530
:	3535 3540
:	Interest charge
:PV :	3455 3490 3515 3520 3535
:	Present value
:QQ :	3480 3515 3520
:	Rate increment
:QV :	9015 9020
:	Used for interest calculations
:X :	3495 3505 9015
:	Loop counter
+-----+	

Total Interest Paid on a Loan Program

```

3450 CLS:PRINT"TOTAL AMOUNT OF INTEREST
PAID ON A LOAN":REM D4
3455 INPUT"Enter amount of loan";PV
3460 INPUT"Total number of payments";N3
3465 INPUT"Number of payments per year";
N2
3470 INPUT"Enter annual interest rate %"
;IR
3475 INPUT"Compounded how many times a y
ear";C
3480 QQ=.1
3485 GOSUB9000:' FIND EFFECTIVE INTEREST
RATE PER PAYMENT
3490 PQ=PV/N3
3495 FOR X=N3 TO 1 STEP -1
3500 P#=(PQ+P#)/(1+(IP/100))
3505 NEXT X
3510 IFINT(P#*100)=PP*100 THEN 3530
3515 IF P#<PV THEN PQ=PQ+(PQ*QQ):P#=0:GO
TO3495
3520 IF P#>PV THEN QQ=QQ*.1:PQ=PQ-(PQ*(Q
Q*10)):P#=0:IF QQ<.0001 THEN 3530 ELSE G
OTO 3495
3525 GOTO3545
3530 PQ=PQ*N3:'TOTAL AMT PAID
3535 PQ=PQ-PV:'TOTAL INTEREST PAID
3540 PRINTUSING"Total interest charge is
$$#,#####.##";PQ
3545 GOSUB10000:RUN3450
9000 IC=(IR/C)/100:'INTEREST RATE PER CO
MPOUNDING PERIOD
9005 CP=C/N2:'COMPOUNDS PER PAYMENT
9010 'FIND EFFECTIVE INTEREST RATE FOR C
2 PERIODS
9015 QV=1:FOR X=1 TO CP:QV=QV+(QV*IC):NE
XT
9020 IP=(QV-1)*100:'IP=INTEREST PER PAYM
ENT
9025 RETURN
10000 PRINT@281,"Press <ENTER> to contin
ue, M for MENU";
10005 A$=INKEY$:IFA$="M"ORA$="m"THENMAXF
ILES=0:CLEARO:MENUELSEIFA$<>CHR$(13)THEN
10005ELSERETURN

```


CONVERTING ADD-ON INTEREST TO ANNUAL INTEREST

Sometimes, instead of using *annual* interest rates, you are faced with *add-on* interest. *Add-on* is actually very simple: just multiply the total amount financed by the add-on rate. This is the amount paid each year for interest. If the loan runs for several years, the interest paid each year is the same. Thus, each payment includes the same fixed amount of money to cover interest.

However, because add-on interest is a fixed amount of each payment, the actual annual interest rate is different. This program takes the financed amount, number of payments, payments per year, and add-on interest rate and calculates what that actually comes to as an annual interest rate.

Examples

<i>Program</i>	<i>Your response</i>
Enter the amount financed?	2000
Total number of payments?	12
Number of payments per year?	12
Enter add-on interest?	6
Interest rate is .920% per period, 11.04% per year, compounded 12 times a year.	
Press <ENTER> to continue, M for Menu	
Enter the amount financed?	4000
Total number of payments?	24
Number of payments per year?	4
Enter add-on interest?	5
Interest rate is 2.220% per period, 8.88% per year, compounded 4 times a year.	
Press <ENTER> to continue, M for Menu	

Line Number and Variable Cross-reference

Ref	Line Numbers
+-----+	
:03550:	3630
:03595:	3605
:03610:	3595
:03630:	3620
:03635:	3590 3605
:10000:	3630
:AR :	3575 3580
:	Add-on interest
:IA :	3570 3575
:	Add-on interest rate
:IC :	3585 3600 3610 3615
:	Interest rate/compounding period
:IP :	3585 3600 3610 3615 3620
:	3625 3640
:	Interest rate per payment
:N2 :	3565 3575 3625
:	Number of payments per year
:N3 :	3560 3575 3580 3635
:	Total number of payments
:P# :	3585 3595 3605 3640
:	Miscellaneous variable
:PQ :	3580 3640
:	Miscellaneous variable
:PV :	3555 3575 3580 3595
:	Present value
:T :	3585 3610
:	Loop counter
:X :	3635 3645
:	Loop counter
+-----+	

Converting Add-on Interest to Annual Interest Program

```

3550 CLS:PRINT" CONVERT ADD-ON INTEREST
TO ANNUAL RATE":REM D5
3555 INPUT"Enter the amount financed";PV
3560 INPUT"Total number of payments";N3
3565 INPUT"Number of payments per year";
N2
3570 INPUT"Enter add-on interest";IA
3575 AR=PV*(IA/100)*(N3/N2):'ADD ON INTE
REST
3580 PQ=(PV+AR)/N3:'PQ=MONTHLY PAYMENTS
3585 IP=0:IC=1:FORT=1 TO 3:P#=1
3590 GOSUB3635
3595 IFP#<PV THEN 3610
3600 IP=IP+IC
3605 P#=1:GOSUB 3635:GOTO 3595
3610 IP=IP-IC:IC=IC/10:NEXT T
3615 IC=IC*10:IP=IP+IC
3620 IF IP<0 THEN PRINT"DATA ERROR. PLEA
SE CHECK YOUR VALUES.":GOTO3630
3625 PRINTUSING"Interest rate is ###.###
% per period,   ###.##% per year, compou
nded ### times ayear.";IP;IP*N2;N2
3630 GOSUB10000:RUN3550
3635 FOR X=N3TO 1 STEP -1
3640 P#=(PQ+P#)/(1+(IP/100)):'MOVE BACK
ONE PMT
3645 NEXT X:RETURN
10000 PRINT@281,"Press <ENTER> to contin
ue, M for MENU";
10005 A$=INKEY$:IF A$="M"OR A$="m"THEN MAXF
ILES=0:CLEARO:MENUELSE IF A$<>CHR$(13)THEN
10005ELSERETURN

```


LOAN REPAYMENT AND REBATE

In the world of high finance, you meet all sorts of people. One of these is Jack the Shark, a lender Mark once met. The last we heard, business for Jack was booming, his loan policy being simple:

"Ya borrow a hunnert dollars now.

"Ya pay me a hunnert next month.

"Ya pay me another hunnert next month, an' we're even."

That worked until an innocent named Lee P. Slicker wanted to pay off his loan after only three weeks. Because he was using Jack's money for a shorter period of time, he figured he needed to repay a lesser amount.

Lee gave Jack \$150. Fortunately, Jack had a payment rule to cover this situation. He called it the *Knuckle Sandwich Rule*.

Banks and lending institutions have a different method of getting you to take a beating if you want to pay off your loan early, called the *Rule of Seventy-eights*.

Every businessperson is likely to reach a point at which he or she must consider the advantages and disadvantages of paying off a loan early. This program takes the basic information about a loan (the original loan amount, annual interest rate, original number of loan payments, and number of payments made so far and their amount) and uses the Rule of Seventy-eights to calculate the amount of money you'll save on interest (an interest rebate) and the amount of money needed to pay off the loan. Add-on annual interest is used.

Examples

<i>Program</i>	<i>Your response</i>
Enter original loan amount?	5000
Enter annual interest rate?	5
Enter original number of loan payments?	36
Number of payments made?	18
Enter amount of payments?	\$159.72

Your interest rebate is: \$172.30

The payoff amount is \$2,702.66.

Press <ENTER> to continue, M for Menu

Enter original loan amount? 6500

Enter annual interest rate? 18

Enter original number of loan
payments? 36

Number of payments made? 25

Enter amount of payments? \$175.00

Your interest rebate is: \$289.86

The payoff amount is \$1,635.14.

Press <ENTER> to continue, M for Menu

Line Number and Variable Cross-reference

Ref	Line Numbers			
+-----+				
:03650:	3710			
:10000:	3710			
:IR	: 3660	3690		
:	Interest rate			
:P0	: 3705	3710		
:	Payoff amount			
:PV	: 3655	3690	3695	
:	Present value			
:RB	: 3695	3700	3705	
:	Rebate			
:X0	: 3665	3680	3690	3695
:	Number of payments			
:X1	: 3670	3675	3680	3695 3705
:	Period			
:X2	: 3685	3705		
:	Amount of payment			
+-----+				

Loan Repayment and Rebate Program

```

3650 CLS:PRINTTAB(7)"LOAN REPAYMENT AND
REBATE":PRINTTAB(12)"BY RULE OF 78'S":RE
M D6
3655 INPUT"Enter original loan amount";P
V
3660 INPUT"Enter annual interest rate";I
R
3665 PRINT"Enter original number of loan
":INPUT"payments";X0
3670 INPUT"Number of payments made";X1
3675 X1=X1+1:' LOAN IS IN THE X1 TH PERI
OD
3680 X1=X0-X1
3685 INPUT"Enter amount of payments";X2
3690 PV=PV*(IR/100)*(X0/12)
3695 RB=PV*(X1+1)*X1/(X0+(X0*X0))
3700 PRINTUSING"Your interest rebate is
$$#,#####.##";RB;:PRINT"."
3705 PO=(RB-(X2*(X1+1)))*-1
3710 PRINTUSING"The payoff amount is $$#
,#####.## .";PO:GOSUB10000:RUN3650
10000 PRINT@281,"Press <ENTER> to contin
ue, M for MENU";
10005 A$=INKEY$:IFA$="M"ORAS$="m"THENMAXF
ILES=0:CLEARO:MENUELSEIFA$<>CHR$(13)THEN
10005ELSERETURN

```


PURCHASE VERSUS LEASE

When acquiring company equipment, you frequently must decide whether it's better to buy the equipment outright or lease it. If you purchase, it's assumed that you will borrow money at current rates, maintain the equipment, and salvage it at the end of its useful lifespan. The lease assumes that you won't be paying the maintenance or getting the salvage.

This program compares the present values of the two alternatives and makes a recommendation as to the best course of action. For example, assume you are interested in acquiring the use of a car, which you may either purchase or lease.

The purchase price is \$5,000. Your down payment would be \$1,000, and you would finance \$4,000 with a 36-month loan, on which you would pay interest of 12 percent per year, compounded monthly. Maintenance would cost \$15 per month. The salvage value would be \$2,500.

On the other hand, a 36-month lease that includes maintenance would cost you \$115 per month.

Assume that banks are currently paying 5 percent annual interest.

Would it be better to lease the car, or buy it?

Example

<i>Program</i>	<i>Your response</i>
Amount to be loaned for purchase (not including down payment)?	4000
Total number of payments?	36
Number of payments per year?	12
Enter annual interest rate (%) charged on loans?	12
Enter annual interest rate (%) you can earn?	5
Compounded how many times a year?	12
Enter down payment?	1000
Enter resale or salvage value?	2500
Maintenance costs per month?	15
Enter lease monthly payments?	115

Loan payments are \$132.85

Present values are:

Purchase value is \$3,780.77

Lease value is \$3,853.04

Recommendation: Purchase it.

Press <ENTER> to continue, M for Menu

Line Number and Variable Cross-reference

Ref	Line Numbers				
+-----+					
:03750:	3995				
:03825:	3845	3850			
:03855:	3840				
:03860:	3850				
:09000:	3815	3860			
:10000:	3995				
:C	: 3785	3875	9000	9005	
:	Compounding frequency				
:CP	: 9005	9015			
:	Compound periods per payment				
:DN	: 3790	3900			
:	Down payment				
:FV	: 3910	3920	3930		
:	Future value				
:IC	: 9000	9015			
:	Interest rate/compound period				
:IE	: 3780	3875			
:	Bank interest rate				
:IP	: 3830	3875	3890	3920	3965
:	9020				
:	Interest per payment				
:IR	: 3775	9000			
:	Annual loan interest rate				
:LE	: 3805	3950			
:	Lease payments				
:MA	: 3800	3870			
:	Maintenance costs				
:N2	: 3770	9005			
:	Number of payments per year				
:N3	: 3765	3820	3825	3885	3915
:	3960				
:	Total number of payments				
:P#	: 3830	3840	3845	3850	3880
:	3890 3900 3955 3965 3975				
:	Used in principal calculations				
:PQ	: 3820	3830	3845	3850	3865
:	3870 3890 3950 3965 3975				
:	Miscellaneous variable				
:PV	: 3760	3820	3840	3845	3850
:	3900 3905 3930 3935 3975				
:	3980 3990				


```
:      Present value, amount borrowed
:QQ   : 3810   3845   3850
:      Miscellaneous variable
:QV   : 9015   9020
:      Effective interest rate
:SV   : 3795   3910
:      Salvage value
:U$   : 3755   3945   3980
:      PRINTUSING format field
:X    : 3825   3835   3885   3895   3915
:      3925   3960   3970
:      9015
:      Loop counter
:X0   : 3905   3935   3945   3990
:      Effective interest rate
+-----+
```


Purchase versus Lease Program

```
3750 CLS:PRINTTAB(6)"LOAN REPAYMENT AND
REBATE":PRINTTAB(12)"BY RULE OF 78'S":PR
INTTAB(11)"PURCHASE OR LEASE":REM D7
3755 US="$ $#,#####.##"
3760 INPUT"Amount to be loaned for purch
ase (not including down payment)";PV
3765 INPUT"Total number of payments";N3
3770 INPUT"Number of payments per year";
N2
3775 INPUT"Enter annual interest rate (%
) charged on loans";IR
3780 INPUT"Enter annual interest rate (%
) you can earn";IE
3785 INPUT"Compounded how many times a y
ear";C
3790 INPUT"Enter down payment";DN
3795 INPUT"Enter resale or salvage value
";SV
3800 INPUT"Maintenance costs per month";
MA
3805 INPUT"Enter lease monthly payments"
;LE
3810 QQ=.1
3815 GOSUB9000:' FIND EFFECTIVE INTEREST
RATE PER PAYMENT
3820 PQ=PV/N3
3825 FOR X=N3 TO 1 STEP -1
3830 P#=(PQ+P#)/(1+(IP/100))
3835 NEXT X
3840 IFINT(P#*100)=PV*100 THEN 3860
3845 IF P#<PV THEN PQ=PQ+(PQ*QQ):P#=0:GO
TO3825
3850 IF P#>PV THEN QQ=QQ*.1:PQ=PQ-(PQ*(Q
Q*10)):P#=0:IF QQ<.0001 THEN 3860 ELSE G
OTO 3825
3860 GOSUB9000
3865 CLS:PRINTUSING"Loan payments are $ $
#,####.##";PQ
3870 PQ=PQ+MA:'ADD MAINTENANCE TO MONTHL
Y LOAD PAYMENTS
3875 IP=IE/C
3880 P#=0
3885 FOR X=N3 TO 1 STEP -1
```



```
3890 P#=(PQ+P#)/(1+(IP/100))
3895 NEXT X
3900 PV=P#+DN
3905 X0=PV
3910 FV=SV
3915 FOR X= 1 TO N3
3920 FV=FV/((IP/100)+1)
3925 NEXT X
3930 PV=FV
3935 X0=X0-PV
3940 PRINT"Present values:"
3945 PRINT"Purchase = ";;PRINTUSINGU$;X0
3950 PQ=LE
3955 P#=0
3960 FOR X=N3-1 TO 1 STEP -1
3965 P#=(PQ+P#)/(1+(IP/100)):'MOVE BACK
ONE PMT
3970 NEXT X
3975 PV=P#+PQ
3980 PRINT"Lease      = ";;PRINTUSINGU$;PV
3985 PRINT"Recommendation: ";
3990 IF X0<PV THEN PRINT"Purchase it." E
LSE PRINT"Lease it.
3995 GOSUB10000:RUN3750
9000 IC=(IR/C)/100:'INTEREST RATE PER CO
MPOUNDING PERIOD
9005 CP=C/N2:'COMPOUNDS PER PAYMENT
9010 'FIND EFFECTIVE INTEREST RATE FOR C
2 PERIODS
9015 QV=1:FOR X=1 TO CP:QV=QV+(QV*IC):NE
XT
9020 IP=(QV-1)*100:'IP=INTEREST PER PAYM
ENT
9025 RETURN
10000 PRINT@281,"Press <ENTER> to contin
ue, M for MENU";
10005 A$=INKEY$:IF A$="M" OR A$="m" THEN MAXF
ILES=0:CLEARO:MENU ELSE IF A$<>CHR$(13) THEN
10005 ELSE RETURN
```


INSTALLMENT PLAN SCHEDULE

Some businesses offer a credit plan in which you're billed a varying amount. The amount charged consists of a fixed amount of the total owed plus the interest on the unpaid balance. Thus, the billings will slowly decline in amount as the interest charged decreases, because the unpaid balance is less each time.

This program takes the cost of the item, the interest charged per billing period, and the number of monthly payments to be made and generates a schedule listing each payment number and amount due.

Examples

<i>Program</i>	<i>Your response</i>
Cost of item?	150
Interest rate as stated on the installment contract % per month on unpaid balance?	1
Number of monthly payments?	8
Print amount of all payments?	Y
Payment #1 = \$20.25	
Payment #2 = \$20.06	
Payment #3 = \$19.88	
Payment #4 = \$19.69	
Payment #5 = \$19.50	
Press <ENTER> to continue, M for Menu	
Payment #6 = \$19.31	
Payment #7 = \$19.13	
Payment #8 = \$18.94	
\$156.75	
Press <ENTER> to continue, M for Menu	
Cost of item?	190
Interest rate as stated on the installment contract % per month on unpaid balance?	1.5
Number of monthly payments?	12

Print amount of all payments?	N
Print amount of which payment?	5
Payment #5 =	\$17.73
Print amount of all payments?	N
Print amount of which payment?	12
Payment #12 =	\$16.07
Print amount of all payments?	Y
Payment #1 =	\$18.68
Payment #2 =	\$18.45
Payment #3 =	\$18.21
Payment #4 =	\$17.97
Payment #5 =	\$17.73
Press <ENTER> to continue, M for Menu	
Payment #6 =	\$17.50
Payment #7 =	\$17.26
Payment #8 =	\$17.02
Payment #9 =	\$16.78
Payment #10 =	\$16.55
Press <ENTER> to continue, M for Menu	
Payment #11 =	\$16.31
Payment #12 =	\$16.07
	<u>\$208.52</u>
Press <ENTER> to continue, M for Menu	

Line Number and Variable Cross-reference

Ref	Line Numbers				
+-----+					
:04000:	4070				
:04035:	4095				
:04040:	4045				
:04050:	4040				
:04075:	4055	4095			
:04095:	4045				
:10000:	4055	4070			
:B\$:	4035	4040			
:	Yes/No				
:N :	4025	4030	4045	4050	4055
:	Number of payments				
:PM :	4040	4045	4055	4080	4085
:	4095				
:	Payment				
:PT :	4055	4065			
:	Total payment				
:X0 :	4075	4080			
:	Miscellaneous variable				
:X4 :	4015	4020	4080		
:	Interest rate				
:X5 :	4030	4075	4080		
:	Miscellaneous variable				
:X6 :	4005	4030	4075		
:	Cost of item				
:X8 :	4080				
:	Miscellaneous variable				
:Z :	4050	4055	4075	4085	4095
:	Loop counter				
+-----+					

Installment Plan Schedule Program

```

4000 CLS:PRINT" Credit Installment Plan
      Schedule":REM d8
4005 INPUT"Cost of item";X6
4010 PRINT"Interest rate as stated on th
e install- ment contract % per month on
unpaid"
4015 INPUT"balance";X4
4020 X4=X4/100
4025 INPUT"Number of monthly payments";N
4030 X5=X6/N
4035 INPUT"Print amount of all payments
(Y/N)";BS
4040 IF LEFT$(B$,1)="n" OR LEFT$(B$,1)="
N" THEN INPUT"Print amount of which paym
ent no.";PM:ELSE 4050
4045 IF PM<1 OR PM>N THEN 4040 ELSE 4095
4050 CLS:FOR Z=1 TO N
4055 GOSUB 4075:PT=PT+PM:IFN>5THENIFINT(
Z/5)=Z/5THENGOSUB10000:CLS
4060 NEXT:PRINT"          -----
- "
4065 PRINT"          ";:PRINT USING"$$#
,#####.##";PT
4070 GOSUB10000:RUN4000
4075 X0=X6-(Z*X5)
4080 X8=X0+X5:PM=X8*X4+X5
4085 PRINT"payment #";Z;"="";:PRINT USING
"$$#,#####.##";PM
4090 RETURN
4095 Z=PM:GOSUB4075:GOTO4035
10000 PRINT@281,"Press <ENTER> to contin
ue, M for MENU";
10005 A$=INKEY$:IFA$="M"ORA$="m"THENMAXF
ILES=0:CLEARO:MENUELSEIFA$<>CHR$(13)THEN
10005ELSERETURN

```


CAPITAL ASSET PRICING MODEL

When dealing with stocks and investments, you must try to determine just how much return on investment you can get. One method of making such a determination is to use the Capital Asset Pricing Model. This model uses the expected rate of return on a portfolio of stocks and securities, a risk-free rate of interest, and a volatility factor to calculate the expected return rate. The volatility factor is a measure of the degree of responsiveness of the stock in question relative to that of the entire portfolio.

This program requires that you enter a risk-free rate of return, the expected rate of return of the portfolio (based on previous performance), and the volatility factor of the individual stock in question.

Examples

<i>Program</i>	<i>Your response</i>
Enter risk-free rate of annual interest (%)?	6
Enter expected rate of return %?	10
Volatility of return?	1.2
Expected rate of return = 10.80%	
using the formula $E(R_i) = R_f + [E(R_m) - R_f]b$	
Where $E(R_i)$ = Expected rate of return	
R_f = Risk-free rate of interest	
$E(R_m)$ = Expected return	
b = Volatility of return	
Press <ENTER> to continue, M for Menu	

Enter risk-free rate of annual
interest (%)? 3
Enter expected rate of return %? 6
Volatility of return? 1.8
Expected rate of return = 8.40%
using the formula $E(R_i) = R_f + [E(R_m) - R_f]b$
Where $E(R_i)$ = Expected rate of return
 R_f = Risk-free rate of interest
 $E(R_m)$ = Expected return
 b = Volatility of return
Press <ENTER> to continue, M for Menu

Line Number and Variable Cross-reference

Ref	Line Numbers
+-----+	
:04100:	4155
:10000:	4155
:B :	4115 4120
:	Volatility factor
:RF :	4105 4120
:	Annual Interest rate
:RI :	4120 4125
:	Expected rate of return
:RM :	4110 4120
:	Expected return percentage
+-----+	

Capital Asset Pricing Model Program

```
4100 CLS:PRINTTAB(7)"CAPITAL ASSET PRICI
NG MODEL":REM D10
4105 PRINT"Enter risk free rate of annua
l":INPUT"interest %";RF
4110 INPUT"Enter expected return %";RM
4115 INPUT"Volatility of return";B
4120 RI=RF+(RM-RF)*B
4125 CLS:PRINTUSING"Expected rate of ret
urn = ###.##,";RI
4130 PRINT "using the formula  $E(R_i)=R_f+[E(R_m)-R_f]b$ "
4135 PRINT "Where  $E(R_i)$ =Expected rate of
return"
4140 PRINTTAB(7)"Rf=Risk-free rate of in
terest"
4145 PRINTTAB(7)"E(Rm)=Expected return"
4150 PRINTTAB(7)"b=Volatility of return"
4155 GOSUB10000:RUN4100
10000 PRINT@281,"Press <ENTER> to contin
ue, M for MENU";
10005 A$=INKEY$:IFA$="M"ORA$="m"THENMAXF
ILES=0:CLEARO:MENUELSEIFA$<>CHR$(13)THEN
10005ELSERETURN
```


CASH MANAGEMENT MODEL

Most business checking accounts don't pay interest, and even a mediocre accountant would rather die than allow large amounts of money to sit idle. The formula used in this program was designed to let you determine the trigger points in your checking account. Reaching the lower limit means you should transfer money from your investment account to the cash account. Reaching the upper limit means you should transfer money out of the cash account into your investment account. To make things more complex, financial institutions usually charge a fee for transferring money from an interest-paying account to a checking account or vice versa, so you have to balance the lost revenues of excess money in the checking account against the cost of transferring additional monies into the account when you need to pay bills. The whole idea is to have your cake and eat it, too.

The program determines these two limits from the bank transferral fee, the minimum cash balance allowed by the bank, the variance of daily changes in the cash balance, and the annual interest rate and number of compounding periods in a year.

Examples

<i>Program</i>	<i>Your response</i>
Enter cost of transferring money from the cash account to the investment account?	25
Enter the minimum cash balance allowed by the bank?	10000
Enter variance of daily changes in the cash balance?	100000
Annual interest rate?	9
How many compounds per year?	365
Should the balance in the cash account reach zero (or the minimum balance) allowed by the bank, transfer \$11,966.45 from the investment account.	

Press <ENTER> to continue, M for Menu
 Transfer money from the cash account to
 the investment account when the balance
 reaches \$15,899.36.

Press <ENTER> to continue, M for Menu

Enter cost of transferring money from the cash account to the investment account?	50
Enter the minimum cash balance allowed by the bank?	5000
Enter variance of daily changes in the cash balance?	200000
Annual interest rate?	10.91
How many compounds per year?	365
Should the balance in the cash account reach zero (or the minimum balance) allowed by the bank, transfer \$5,630.73 from the investment account.	

Press <ENTER> to continue, M for Menu
 Transfer money from the cash account to
 the investment account when the balance
 reaches \$6,892.19.

Press <ENTER> to continue, M for Menu

Line Number and Variable Cross-reference

Ref	Line Numbers
+-----+	
:04200:	4280
:10000:	4265 4280
:B :	4210 4240
:	Money transference cost
:C :	4230 4235
:	Compound periods per year
:I :	4235 4240
:	Interest rate/compounding period
:IA :	4225 4235
:	Annual interest rate
:MB :	4215 4255 4275
:	Minimum cash balance
:Q2 :	4220 4240
:	Variance
:RP :	4240 4245 4255 4275
:	Transfer amount
+-----+	

Cash Management Model Program

```
4200 CLS:PRINTTAB(7)"CASH MANAGEMENT MOD
EL":REM d11
4205 PRINT"Enter cost of transferring mo
ney from the cash account to the inves
tment"
4210 INPUT"account";B
4215 INPUT"Enter the minimum cash balanc
e allowed by the bank";MB
4220 INPUT"Enter variance of daily chang
es in the cash balance";Q2
4225 INPUT"Annual interest rate";IA
4230 INPUT"How many compounds per year";
C
4235 I=(IA/C)/100
4240 RP=3*B*Q2/(4*I)
4245 RP=RP^(1/3)
4250 CLS:PRINT"Should the balance in the
cash account reach zero (or the minimu
n balance) allowed by the bank, tran
sfer"
4255 PRINTUSING"$$#,#####.##";RP+MB;
4260 PRINT" from the investment":PRINT"a
ccount."
4265 GOSUB10000:CLS
4270 PRINT"Transfer money from the cash
account to the investment account when t
he balance reaches ";
4275 PRINT USING"$$#,#####.##";(RP*3)+
MB
4280 GOSUB10000:RUN4200
10000 PRINT@281,"Press <ENTER> to contin
ue, M for MENU";
10005 A$=INKEY$:IF A$="M"OR A$="m"THEN MAXF
ILES=0:CLEARO:MENUELSEIF A$<>CHR$(13)THEN
10005ELSERETURN
```


PRICE BEFORE SALES TAX

This program uses the total price paid for an item and the sales tax rate to calculate the original retail price. In other words, if you purchased an item for \$1,000, tax included, what did the item cost before tax?

Examples

<i>Program</i>	<i>Your response</i>
Enter total price paid?	1000
Enter sales tax?	7
Purchase price without sales tax is \$934.58	
Total tax is \$65.42	
Press <ENTER> to continue, M for Menu	
Enter total price paid?	2000
Enter sales tax?	10
Purchase price without sales tax is \$1,818.18	
Total tax is \$181.82	
Press <ENTER> to continue, M for Menu	

Line Number and Variable Cross-reference

Ref	Line Numbers		
+-----+			
:04300:	4330		
:10000:	4330		
:PC :	4305	4315	4325
:	Total price		
:SP :	4315	4320	4325
:	Price without sales tax		
:ST :	4310	4315	
:	Sales tax		
+-----+			

Price before Sales Tax Program

```
4300 CLS:PRINTTAB(7)"PRICE BEFORE SALES  
TAX":REM m2  
4305 INPUT"Enter total price paid";PC  
4310 INPUT"Enter sales tax";ST  
4315 SP=PC/(1+(ST/100))  
4320 PRINTUSING"Purchase price, without  
sales tax is:    $$#,#####.##";SP  
4325 PRINTUSING"Total tax is $$#,#####.##"  
#";PC-SP  
4330 GOSUB10000:RUN4300  
10000 PRINT@281,"Press <ENTER> to contin  
ue, M for MENU";  
10005 A$=INKEY$:IFA$="M"ORA$="m"THENMAXF  
ILES=0:CLEARO:MENUELSEIFA$<>CHR$(13)THEN  
10005ELSERETURN
```


MARKUP (CALCULATING RETAIL PRICE)

Most items are sold on a cost-plus basis. A certain percentage, or *markup*, is applied to the wholesale price, or *cost*, to determine the selling price.

If you paid \$2,000 wholesale for the item and have a policy of marking up your products by 45 percent, what is the selling price?

Examples

<i>Program</i>	<i>Your response</i>
Enter cost of good?	2000
Enter markup %?	45
Selling price is \$2,900.00	
Press <ENTER> to continue, M for Menu	
Enter cost of good?	1000
Enter markup %?	100
Selling price is \$2,000.00	
Press <ENTER> to continue, M for Menu	

Line Number and Variable Cross-reference

Ref	Line Numbers
+-----+	
:04350:	4370
:10000:	4370
:C :	4355 4365
:	Cost of goods
:M :	4360 4365
:	Percent markup
+-----+	

Markup (Calculating Retail Price) Program

```
4350 CLS:PRINTTAB(15)"MARKUPS":PRINT:REM
      M3
4355 INPUT"Enter cost of goods";C
4360 INPUT"Enter markup %";M
4365 PRINTUSING"Selling price is $$#,###
      ##.##";C+(C*(M/100))
4370 GOSUB10000:RUN4350
10000 PRINT@281,"Press <ENTER> to contin
      ue, M for MENU";
10005 A$=INKEY$:IFA$="M"ORA$="m"THENMAXF
      ILES=0:CLEARO:MENUELSEIFA$<>CHR$(13)THEN
      10005ELSERETURN
```


COST BEFORE MARKUP

If you know the selling price and the markup, you can calculate the wholesale cost. If a competitor is selling your \$500 computer for \$200 and her markup is 45 percent, how much is she buying them for?

Examples

<i>Program</i>	<i>Your response</i>
Enter sales price?	200
Enter seller's markup?	45
Seller's cost is \$137.93	
Press <ENTER> to continue, M for Menu	
Enter sales price?	2000
Enter seller's markup %?	50
Seller's cost is \$1,333.33	
Press <ENTER> to continue, M for Menu	

Line Number and Variable Cross-reference

Ref		
+-----+		
:04375:	4395	
:10000:	4395	
:M :	4385	4390
:	Markup	
:SP :	4380	4390
:	Sales Price	
+-----+		

Line Numbers

Cost before Markup Program

```
4375 CLS:PRINTTAB(10)"COST BEFORE MARKUP
":REM M4
4380 PRINT:INPUT"Enter sales price";SP
4385 INPUT"Enter seller's markup %";M
4390 PRINTUSING"Seller's cost is $$#,###
###.##";SP/(1+(M/100))
4395 GOSUB10000:RUN4375
10000 PRINT@281,"Press <ENTER> to contin
ue, M for MENU";
10005 A$=INKEY$:IFA$="M"ORA$="m"THENMAXF
ILES=0:CLEARO:MENUELSEIFA$<>CHR$(13)THEN
10005ELSERETURN
```


PERCENTAGE MARKUP

This program takes the cost and selling price of an item and calculates the percentage of markup over cost and percentage of retail price.

Examples

<i>Program</i>	<i>Your response</i>
Enter cost?	2000
Enter selling price?	3000
Markup of cost is 50.00%	
Profit margin is 33.33%	
Press <ENTER> to continue, M for Menu	
Enter cost?	1000
Enter selling price?	1500
Markup of cost is 50.00%	
Profit margin is 33.33%	
Press <ENTER> to continue, M for Menu	
Enter cost?	1000
Enter selling price?	1250
Markup cost is 25.00%	
Profit margin is 20.00%	
Press <ENTER> to continue, M for Menu	

Line Number and Variable Cross-reference

Ref	Line Numbers			
+-----+				
:04400:	4435			
:10000:	4435			
:C	: 4405	4415	4425	
:	Cost			
:M	: 4415	4420	4425	4430
:	Markup			
:SP	: 4410	4415	4425	
:	Selling price			
+-----+				

Percentage Markup Program

```
4400 CLS:PRINTTAB(15)"PERCENT MARKUP":RE
M M5
4405 PRINT:INPUT"Enter cost";C
4410 INPUT"Enter selling price";SP
4415 M=(SP/C)-1
4420 PRINTUSING"Markup of cost is ###.##
%";M*100
4425 M=-1*((C/SP)-1)
4430 PRINTUSING"Profit margin is ###.##%
";M*100
4435 GOSUB10000:RUN4400
10000 PRINT@281,"Press <ENTER> to contin
ue, M for MENU";
10005 A$=INKEY$:IFA$="M"ORA$="m"THENMAXF
ILES=0:CLEARO:MENUELSEIFA$<>CHR$(13)THEN
10005ELSERETURN
```


DISCOUNTED PRICE

If an item is discounted 30 percent and it normally sells for \$2,000, what will you pay for it?

Examples

<i>Program</i>	<i>Your response</i>
Enter regular price?	2000
Enter discount %?	30
Sale price is \$1,400.00 Press <ENTER> to continue, M for Menu	
Enter regular price?	3000
Enter discount %?	75
Sale price is \$750.00 Press <ENTER> to continue, M for Menu	

Line Number and Variable Cross-reference

Ref	Line Numbers
+-----+	
:04450:	4470
:10000:	4470
:D :	4460 4465
:	Discount
:P :	4455 4465
:	Price
+-----+	

Discounted Price Program

```
4450 CLS:PRINTTAB(10)"THE DISCOUNTED PRI
CE":REM M6
4455 PRINT:INPUT"Enter regular price";P
4460 INPUT"Enter discount %";D
4465 PRINTUSING"Sale price is $$#,#####
.##";P-(P*(D/100))
4470 GOSUB10000:RUN4450
10000 PRINT@281,"Press <ENTER> to contin
ue, M for MENU";
10005 A$=INKEY$:IFA$="M"ORA$="m"THENMAXF
ILES=0:CLEARO:MENUELSEIFA$<>CHR$(13)THEN
10005ELSERETURN
```

PRICE BEFORE DISCOUNT

If an item is priced at \$2,000, including a 45 percent discount, what is the regular price?

Examples

<i>Program</i>	<i>Your response</i>
Enter discounted price?	2000
Enter discount %?	45
Regular price is \$3,636.36	
Press <ENTER> to continue, M for Menu	
Enter discounted price?	1500
Enter discount %?	50
Regular price is \$3,000.00	
Press <ENTER> to continue, M for Menu	

Line Number and Variable Cross-reference

Ref	Line Numbers
+-----+	
:04500:	4520
:10000:	4520
:D	: 4510 4515
:	Discount
:P	: 4505 4515
:	Price
+-----+	

Price before Discount Program

```
4500 CLS:PRINTTAB(10)"PRICE BEFORE DISCO
UNT":REM M8
4505 PRINT:INPUT"Enter discounted price"
:P
4510 INPUT"Enter discount %";D
4515 PRINTUSING"Regular price is $$#,###
###.##";P/(1-(D/100))
4520 GOSUB10000:RUN4500
10000 PRINT@281,"Press <ENTER> to contin
ue, M for MENU";
10005 A$=INKEY$:IFA$="M"ORA$="m"THENMAXF
ILES=0:CLEARO:MENUELSEIFA$<>CHR$(13)THEN
10005ELSERETURN
```


PROFIT MARGIN

If you buy a product for resale at \$1,000, and want to have made a 45 percent profit after selling it, what retail price should it have? This program calculates selling price using desired profit.

Examples

<i>Program</i>	<i>Your response</i>
Enter cost of good?	1000
Enter desired profit margin %?	45
Selling price is \$1,818.18	
Press <ENTER> to continue, M for Menu	
Enter cost of good?	2000
Enter desired profit margin %?	50
Selling price is \$4,000.00	
Press <ENTER> to continue, M for Menu	

Line Number and Variable Cross-reference

Ref	Line Numbers
+-----+	
:04550:	4575
:10000:	4575
:C :	4555 4565
:	Cost of item
:M :	4560 4565
:	Margin
:SP :	4565 4570
:	Selling price
+-----+	

Profit Margin Program

```
4550 CLS:PRINTTAB(15)"PROFIT MARGIN":REM
      M9
4555 PRINT:INPUT"Enter cost of good";C
4560 INPUT"Enter desired profit margin %
      ";M
4565 SP=(C/(1-(M/100)))
4570 PRINTUSING"Selling price is $$#,###
      ###.##";SP
4575 GOSUB10000:RUN4550
10000 PRINT@281,"Press <ENTER> to contin
      ue, M for MENU";
10005 A$=INKEY$:IFA$="M"ORA$="m"THENMAXF
      ILES=0:CLEARO:MENUELSEIFA$<>CHR$(13)THEN
10005ELSERETURN
```


PIE CHART

This program draws a pie chart on the Model 100 display. First the program asks you how many data items you have (so it will know into how many pieces to divide the pie), and then it asks for the data items, one at a time. When you're finished entering the data, the program adds the numbers together to get the total. Then it calculates the percentage of the total pie each individual piece gets. With this information, the program draws a circle on the left side of your display and divides it into slices. Each slice is labeled sequentially starting with the letter *a* and progressing toward *z*. On the right side of the display, each data item is placed beside its corresponding label, and its percentage of the total pie is listed.

When you're finished with the chart, just press <ENTER> and the program will return you to BASIC. This is the only program that doesn't automatically rerun or return you to the menu.

Because of the various types of printers on the market and the fact that each has its own method of drawing graphics, we can draw the pie chart only on the display.

Line Number and Variable Cross-reference

Ref	Line Numbers				
+-----+					
:04840:	4840	4925			
:04850:	4815	4825			
:04855:	4830				
:04885:	4835				
:04900:	4925				
:04925:	4890				
:04935:	4950				
:04940:	4935	4945			
:04945:	4815				
:04950:	4950				
:04955:	4815				
:A	: 4805	4855	4860		
:	Calculation variable				
:A\$: 4840	4870	4875		
:	Keyboard input				
:BX	: 4935				
:	X-variable				
:BY	: 4935				
:	Y-variable				
:DA(: 4825	4895	4900	4965	4975
:	4980				
:	Data item				
:FM\$: 4885	4900			
:	PRINTUSING format field				
:I	: 4820	4825	4830	4870	4895
:	4900	4910	4915	4970	4975
:	4980				
:	Alphabetic character loop				
:P2	: 4810	4825	4830	4950	
:	Value for PI				
:W	: 4895	4900			
:	Percent of total				
:X	: 4845	4850	4860	4865	4935
:	4940				
:	Calculation variable (x-axis)				
:X1	: 4845	4850	4860	4865	4935
:	4940				
:	Calculation variable (y-axis)				
:X4	: 4815	4855			


```

:      Calculation variable
:X5   : 4845   4945   4950
:      Calculation variable
:X6   : 4825   4895   4970   4980
:      Total data
:X7   : 4815   4825   4830   4845   4850
:      4855   4940   4945   4950
:      Calculation variable
:X9   : 4865   4875   4885   4900   4905
:      4910
:      Print location
:Z    : 4805   4820   4895   4955   4965
:      4970
:      Number of data items
+-----+

```


Pie Chart Program

```

4800 CLS: CLEAR 500: PRINT TAB(15) "PIE CHART
": REM pie
4805 DEFSNGA-Z
4810 P2=3.1416*2
4815 GOSUB 4955: GOSUB 4945: X7=0: GOSUB 4850
: X4=0
4820 FOR I=1 TO Z-1
4825 X7=X7+P2*DA(I)/X6: GOSUB 4850
4830 GOSUB 4855: NEXT I: X7=P2: GOSUB 4855
4835 GOSUB 4885
4840 A$=INKEY$: IF A$="" THEN 4840: ELSE END
4845 X=X5*SIN(X7)+45: X1=X5*COS(X7)*.9+32
: PSET(X,X1): RETURN
4850 X=24*SIN(X7)+45: X1=24*COS(X7)*.9+32
: LINE(45,32)-(X,X1): RETURN
4855 A=X4+(X7-X4)/2: X4=X7
4860 X=33*SIN(A)+45: X1=33*COS(A)*.9+32
4865 X9=(X1\8)*40+(X\6)
4870 A$=CHR$(96+I)
4875 PRINT@X9,A$;
4880 RETURN
4885 X9=16: FM$="### ##.##"
4890 ON ERROR GOTO 4925
4895 FOR I=1 TO Z: W=DA(I)/X6*100.
4900 PRINT@X9,USING FM$;CHR$(96+I);DA(I)
;W;
4905 X9=X9+12
4910 IF I MOD 2=0 THEN X9=X9+16
4915 NEXT I
4920 RETURN
4925 IF ERL=4900 THEN RESUME 4840
4930 ON ERROR GOTO 0: RESUME
4935 BX=X: BY=X1: GOSUB 4940: LINE(BX,BY)-(X
,X1): RETURN
4940 X=24*SIN(X7)+45: X1=24*COS(X7)*.9+32
: RETURN
4945 X5=24: X7=0: GOSUB 4940
4950 GOSUB 4935: X7=X7+.3: IF X7<P2 THEN 4
950: ELSE X5=24: X7=0: GOSUB 4935: RETURN
4955 INPUT "How many data items do you ha
ve"; Z
4965 DIM DA(Z)

```



```
4970 X6=0:FOR I=1TOZ
4975 PRINT"#";I;:INPUTDA(I)
4980 X6=X6+DA(I):NEXTI
4985 CLS:RETURN
```


ECONOMIC ORDER QUANTITY

One of the figures a businessperson must juggle is *inventory level*. It costs money to keep stock on hand; yet, the more units you buy, the lower the per-unit cost. This program takes the cost of placing an order, the units needed per period of time, and the cost of storage for the same amount of time and then calculates the most efficient quantity for you to order each time you reorder.

Example

<i>Program</i>	<i>Your response</i>
Cost of placing an order?	25
Number of units needed per period?	1000
Per-period storage, warehousing, or carrying cost per unit (if any)?	2
Your order should be 158	
Press <ENTER> to continue, M for Menu	

Line Number and Variable Cross-reference

Ref	Line Numbers
+-----+	
:05000:	5040
:10000:	5040
:EQ :	5030 5035
:	Units to order
:X1 :	5005 5020
:	Cost of placing order
:X2 :	5010 5020
:	Number of units needed/period
:X3 :	5015 5025 5030
:	Storage cost per unit per period
:X4 :	5020 5030
:	Miscellaneous variable
+-----+	

Economic Order Quantity Program

```
5000 CLS: CLEAR 500: PRINT TAB(10) "ECONOMIC  
ORDER QUANTITY": REM EQQ  
5005 INPUT "Cost of placing an order"; X1  
5010 INPUT "Number of units needed per pe  
riod"; X2  
5015 INPUT "Per period storage, warehousi  
ng, or carrying costs per unit (if a  
ny)"; X3  
5020 X4 = 2 * X1 * X2  
5025 IF X3 = 0 THEN X3 = .000001  
5030 EQ = SQR(X4 / X3)  
5035 PRINT "Your order should be"; INT(EQ)  
: PRINT "units."  
5040 GOSUB 10000: RUN 5000  
10000 PRINT @281, "Press <ENTER> to contin  
ue, M for MENU";  
10005 A$ = INKEY$: IF A$ = "M" OR A$ = "m" THEN MAXF  
ILES = 0: CLEAR 0: MENU ELSE IF A$ <> CHR$(13) THEN  
10005 ELSE RETURN
```


BAR CHART

This program draws a bar chart on the basis of data you give it and stores the chart in RAM for your use in a text file or for printing out. Depending on whether you intend to print out the bar chart or include it in a text file, the type of bar chart is altered. If you're going to print it out, exclamation marks and dashes are used to delineate the chart boundaries. If you're just going to view it in RAM, then graphic symbols are used. *Note:* The graphic symbols wouldn't print on our printer, which substituted the letters *z*, *o*, *u*, and *q* instead. When you type in line 5295, which defines the graphic symbols, instead of pressing the lowercase *z*, press <GRPH><SHIFT>K. In place of the *o*, press <GRPH><SHIFT>C. For the lowercase *u*, use <GRPH><SHIFT>:. And for the *q*, use <GRPH><SHIFT>P.

The program asks you for the maximum value of the entries (the highest value of any of them), the increment, what the bottom line of the graph should read (the limit is 30 characters, where each character will have one value assigned to it), the name of the bar chart (limited to 9 characters), and then the values for each of the data items. When the last item is entered, the chart's filename is requested, and then the chart is drawn in RAM.

In the example below, the chart is named *Sales/mth* (sales per month), and the bottom line of the chart reads:

J F M A M J J

The letters stand for the initials of the months January, February, March, April, May, June, and July. Each letter is followed by three blank spaces, representing three blank weeks. Thus, each monthly period is usually displayed as four weeks, each of which is prompted for its value by the program:

<i>Program</i>	<i>Your response</i>
Value for J	20
Value for	23
Value for	25
Value for	30
Value for F	40
Value for	45
Value for	50
Value for	50

Value for M	55
Value for	60
Value for	60
Value for	60
Value for A	70
Value for	70
Value for	80
Value for	70
Value for M	60
Value for	50
Value for	40
Value for	45
Value for J	50
Value for	60
Value for	70
Value for	80
Value for J	90
Value for	95
Value for	90
Value for	90

When you've finished entering the data, type the filename under which you want the chart stored. When the *Press <ENTER>* to continue, *M* for *MENU* prompt appears, press the M key, and return to the menu. Now put the cursor over the chart's filename, and take a look at it. For the example I've used, the file looks like this:

```

100!
 90!                                XXXX
 80!                                X      XXXXX
 70!                                XXXX   XXXXXX
 60!                                XXXXXXXX XXXXXXXX
 50!                                XXXXXXXXXXXX XXXXXXXX
 40!                                XXXXXXXXXXXXXXXXXXXXXXXX
 30!                                XXXXXXXXXXXXXXXXXXXXXXXX
 20! XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
 10! XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

```

Sales/mth! J F M A M J J

Value for J 20
 Value for 23
 Value for 25
 Value for 30
 Value for F 40
 Value for 45
 Value for 50
 Value for 50
 Value for M 55
 Value for 60
 Value for 60
 Value for 60
 Value for A 70
 Value for 70
 Value for 80
 Value for 70
 Value for M 60
 Value for 50
 Value for 40
 Value for 45
 Value for J 50
 Value for 60
 Value for 70
 Value for 80
 Value for J 90
 Value for 95
 Value for 90
 Value for 90

Line Number and Variable Cross-reference

Ref	Line Numbers				
+-----+					
:05050:	5285				
:05070:	5060	5075			
:05125:	5135				
:05140:	5145				
:05160:	5175				
:05200:	5180				
:05290:	5065				
:05295:	5060	5065			
:10000:	5285				
:BL\$:	5190	5290	5295		
:	Print character				
:D\$(:	5095	5105	5110	5120	5150
:	5190	5240	5245		
:	Columns of graph				
:D(:	5095	5170	5175	5180	5185
:	5270				
:	Value for above				
:DA\$:	5120	5290	5295		
:	Print character				
:I\$:	5060	5065			
:	Input character				
:IC :	5085	5090	5105	5180	5185
:	Increment				
:L :	5180				
:	Calculation variable				
:LI\$:	5110	5150	5290	5295	
:	Print character				
:MA :	5070	5075	5090	5175	
:	Maximum value				
:MI :	5080	5090	5175		
:	Minimum value				
:N :	5090	5095	5100	5105	5190
:	5235				
:	Number of columns				
:PL\$:	5120	5290	5295		
:	Print character				
:X :	5100	5105	5110	5120	5150
:	5155	5165	5170	5175	5180
:	5185	5190	5200	5235	5240
:	5245	5250	5255	5265	5270

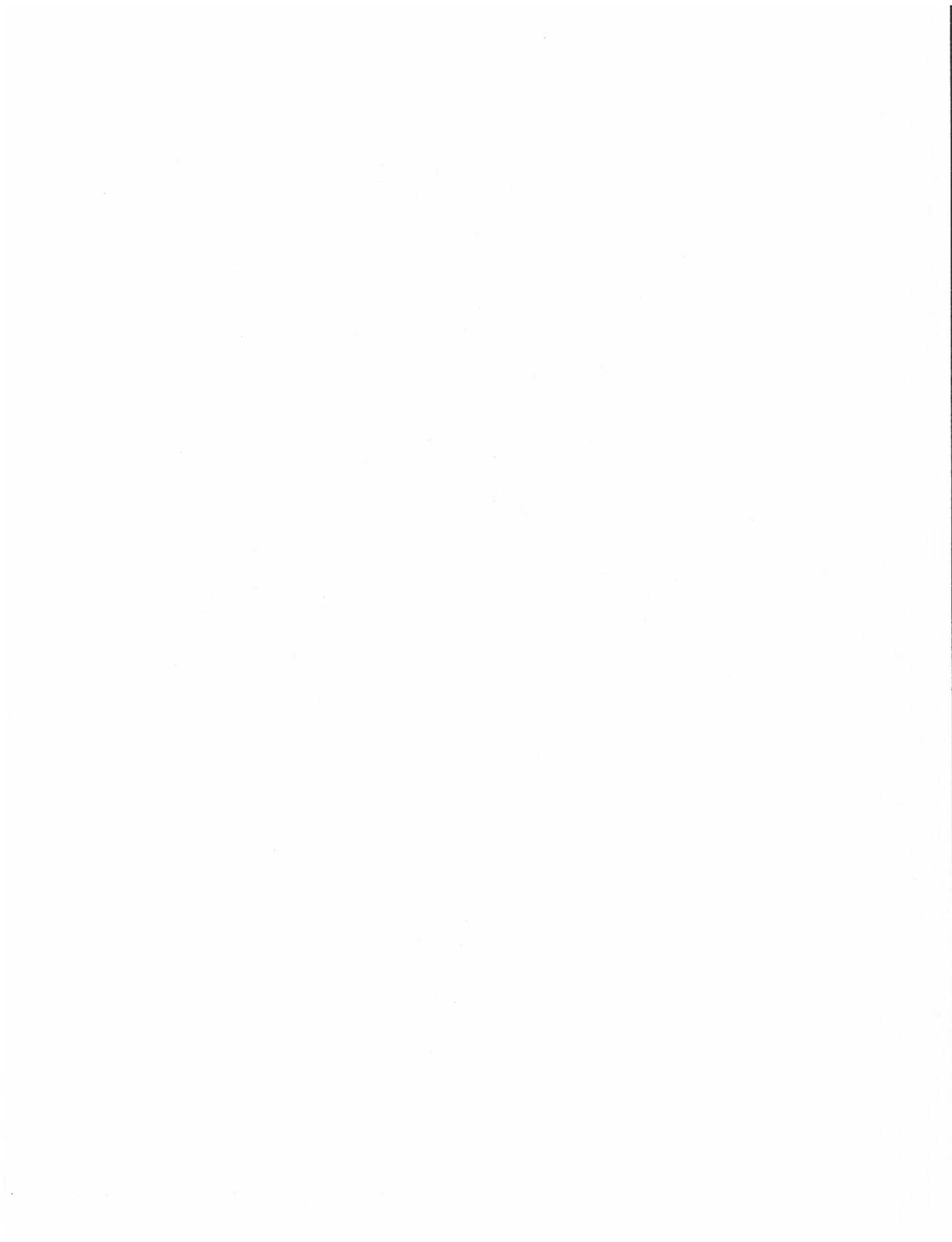

```
:      5275
:      Loop counter
:X$    : 5135    5150    5155    5165
:      5255    5265
:      Bottom line of graph
:Y$    : 5145    5150
:      Data description
:Z     : 5185    5190    5195
:      Counter
+-----+
```

Bar Chart Program

```
5050 CLS:PRINTTAB(15)"BAR CHART":REM BAR
5055 CLEAR 2000
5060 INPUT"Do you intend to make a print
out (Y/N)";I$:IFI$="N"ORI$="n"THENGOSUB5
295:GOTO5070
5065 INPUT"Will your printer print graph
ics (Y/N)";I$:IFI$="N"ORI$="n"THENGOSUB5
290 ELSE GOSUB5295
5070 INPUT"Maximum value of entries";MA
5075 IFMA>9999999 THENPRINT"value too hi
gh":GOTO5070
5080 MI=0
5085 INPUT"Increment for graph      ";IC
5090 N=INT((MA-MI)/IC)
5095 DIMD$(N+2),D(30)
5100 FORX=0TON-1
5105 D$(X)=STR$((N-X)*IC)
5110 D$(X)=STRING$(9-LEN(D$(X))), " ")+D
$(X)+LI$+STRING$(29," ")
5115 NEXT
5120 D$(X)=STRING$(9,DA$)+PL$+STRING$(29
,DA$)
5125 CLS
5130 PRINT"Bottom line to read (30 chrs.
max):"
5135 INPUTX$:IFLEN(X$)>30THEN GOTO 5125
5140 PRINT"Data description (9 chrs. max
):"
5145 INPUTY$:IFLEN(Y$)>9THEN5140
5150 D$(X+1)=Y$+STRING$(9-LEN(Y$)," ")+L
I$+X$
5155 FORX=1TOLEN(X$)
5160 PRINT"Value for ";
5165 PRINTMID$(X$,X,1);
5170 INPUTD(X)
5175 IFD(X)>MAORD(X)<MITHENPRINT"Redo.":
GOTO5160
5180 L=INT(D(X)/IC):IFL=0THEN5200
5185 FORZ=1TOINT(D(X)/IC)
5190 MID$(D$(N-Z),X+10,1)=BL$
5195 NEXTZ
5200 NEXTX
5205 INPUT"Filename of bar chart";FN$
5210 IF FN$=""THENFN$="bar"
```



```
5215 FN$=LEFT$(FN$,6)
5220 FN$="RAM:"+FN$+".do"
5225 PRINT"Chart will be saved as ";FN$
5230 OPEN FN$ FOR OUTPUT AS 1
5235 FORX=0TON+1
5240 PRINT#1,D$(X)
5245 PRINTD$(X)
5250 NEXTX
5255 FORX=1TOLEN(X$)
5260 PRINT#1,"Value for ";
5265 PRINT#1,MID$(X$,X,1);
5270 PRINT#1,D(X)
5275 NEXTX
5280 CLOSE
5285 GOSUB10000:RUN5050
5290 PL$="+":BL$="X":LI$="!":DA$="-":RET
URN
5295 PL$="z":BL$="o":LI$="u":DA$="q":RET
URN
10000 PRINT@281,"Press <ENTER> to contin
ue, M for MENU";
10005 A$=INKEY$:IFA$="M"ORAS$="m"THENMAXF
ILES=0:CLEARO:MENUELSEIFA$<>CHR$(13)THEN
10005ELSERETURN
```

3

Accountant's Helpers

The accountant has been described as the single most valuable professional in society today; in his or her clear understanding of the interrelationship of economic factors and solid grasp of the hyper-complex tax system, the accountant is said to be unmatched by any other professional. An accountant has also been described as someone who couldn't make it as a bookkeeper. Regardless, the programs in this chapter are some of the accountant's tools.

A whole field of accounting is dedicated to allocating costs fairly. If you make a product, you need to know exactly how much it costs to make. This allows you to price the product accurately and make an appropriate profit while being competitive on the open market. If, for example, you purchase a machine to make the product during the first year of your operation, is it fair to say that the product costs more to make in the first year (because of the cost of the machine) and less to make each following year? Of course not. If this were the case, then new businesses could never compete with existing firms. For this reason, we have an accounting tool called *depreciation*.

Depreciation allows a business to spread the cost of an item over several years, or until the machine breaks down and needs to be sold or disposed of. This way, even though your company pays for the machine during the first year, the tremendous burden of paying for it does not need to be passed on to the customers in full right away. It can be spread out over a period equal to that for which the machine is expected to be operable. This is referred to as *depreciation and amortization*.

Other accounting programs presented here deal with earnings per share calculations and investment yield.

DEPRECIATION

Whenever a business buys equipment, the accountant or executive in charge must decide how to handle the decline of its value over the useful lifespan of the item. For low-cost equipment, the best choice is to write off the entire cost as a business expense, subtracting the money from the gross profits and reducing the amount of tax liability the company faces at year's end.

With expensive purchases, many different options are available that let the company spread out the cost of the purchase over the lifespan of the equipment, thus reducing the company's tax liability in several different years.

All three depreciation programs listed here take the cost of the item and subtract the salvage value, which in some cases is little more than the value of the raw materials from which the item was made. This gives the actual cost of the purchase.

This section lists programs that calculate three kinds of depreciation (straight-line, sum-of-year's-digits, and declining) so you can see how each sort affects the tax write-off each year. The fourth program compares straight-line and declining depreciation, telling you which is better for a given item and suggesting when you should switch from one to the other for the best tax advantage.

STRAIGHT-LINE DEPRECIATION

The first program calculates straight-line depreciation. Simply take the purchase cost, subtract the salvage value, and divide by the lifespan of the item, in years.

To use the program, just give it the purchase price, salvage value, lifespan (in years), and purchase month and year. The program will calculate the first year's depreciation (the item may not receive a full year's service), the yearly depreciation, and the final year's depreciation (which may also be less than a full year). The total of the first year's and last year's depreciation should be the same as a full year's depreciation, within a few pennies.

Examples

<i>Program</i>	<i>Your response</i>
Purchase price?	25000
Salvage value?	500
Life in years?	9
Purchase month and year (MM/YY)?	02/80
Depreciation from 02/01/80 to 12/31/80: \$2,495.37	
Depreciation for remaining years: \$2,722.22	
Depreciation from 01/01/89 to 02/01/89: \$226.85	
Press <ENTER> to continue, M for Menu	
Purchase price?	1000
Salvage value?	100
Life in years?	8
Purchase month and year (MM/YY)?	09/83
Depreciation from 09/01/83 to 12/31/83: \$37.50	
Depreciation for remaining years: \$112.50	
Depreciation from 01/01/91 to 09/01/91: \$75.00	
Press <ENTER> to continue, M or Menu	

Line Number and Variable Cross-reference

Ref	Line Numbers
+-----+	
:02500:	2575
:02520:	2520
:02525:	2535 2540
:10000:	2575
:C :	2510 2545
:	Purchase price
:D :	2545 2550 2565 2570
:	Depreciation per year
:D1 :	2555 2560 2570
:	First year's depreciation
:DD\$:	2525 2530
:	Purchase date
:DM :	2550 2555
:	Depreciation per month
:L :	2520 2545 2570
:	Useful lifespan
:MM :	2530 2535 2555 2560 2570
:	Purchase month
:S :	2515 2545
:	Salvage value
:U\$:	2505 2560 2565 2570
:	PRINTUSING format field
:YY :	2530 2540 2560 2570
:	Purchase year
+-----+	

Straight-line Depreciation Program

```

2500 CLS:PRINT"    STRAIGHT LINE DEPRECIA
TION":REM B1
2505 U$="$$#,#####.##"
2510 INPUT"Purchase price";C
2515 INPUT"Salvage value";S
2520 INPUT"Life in years";L:IF L<1 THEN
2520
2525 INPUT"Purchase month and year (MM/Y
Y)";DD$
2530 MM=VAL(LEFT$(DD$,2)):YY=VAL(RIGHT$(
DD$,2))
2535 IFMM<10RMMM>12THEN2525
2540 PRINT:IFY<0THEN2525
2545 D=(C-S)/L:'DEPRECIATION PER YEAR
2550 DM=D/12:'DEPRECIATION PER MONTH
2555 D1=DM*(13-MM)
2560 PRINT"Depreciation from ";RIGHT$(ST
R$(MM),2);"/01/";RIGHT$(STR$(YY),2);" to
12/31/";RIGHT$(STR$(YY),2);":":PRINTUSI
NGU$;D1
2565 PRINT"Depreciation for remaining ye
ars:":PRINTUSINGU$;D
2570 PRINT"Depreciation from 01/01/";RIG
HT$(STR$(YY+L),2);" to ";RIGHT$(STR$(MM)
,2);"/01/";RIGHT$(STR$(YY+L),2);":":PRIN
TUSINGU$;D-D1
2575 GOSUB10000:RUN2500
10000 PRINT@281,"Press <ENTER> to contin
ue, M for MENU";
10005 A$=INKEY$:IFA$="M"ORA$="m"THENMAXF
ILES=0:CLEARO:MENUELSEIFA$<>CHR$(13)THEN
10005ELSERETURN

```


SUM-OF-YEAR'S-DIGITS DEPRECIATION

The sum-of-year's-digits method of depreciation multiplies the depreciable cost of the purchase by the ratio of the number of remaining years to the sum of all the years in the item's lifespan.

To use this program, just type in the lifespan of the asset, its cost, the purchase date, and the final salvage value. From this information, the program will create a display of each year's applicable depreciation. As in the previous program, the first and last years are not full years, but together add up to one year so that the entire time considered is a whole number of years.

Examples

<i>Program</i>	<i>Your response</i>
Life of asset in years?	9
Cost of asset?	25000
Month and year purchased?	02/80
Salvage value?	500
Depreciation from	
02/01/80 to 12/31/80 is	\$4,491.67
01/01/81 to 12/31/81 is	\$4,400.93
01/01/82 to 12/31/82 is	\$3,856.48
01/01/83 to 12/31/83 is	\$3,312.04
01/01/84 to 12/31/84 is	\$2,767.59
01/01/85 to 12/31/85 is	\$2,223.15
Press <ENTER> to continue, M for Menu	
01/01/86 to 12/31/86 is	\$1,678.70
01/01/87 to 12/31/87 is	\$1,134.26
01/01/88 to 12/31/88 is	\$589.81
01/01/89 to 02/01/89 is	\$45.37
Remaining salvage value is	\$500.00
Press <ENTER> to continue, M for Menu	

Life of asset in years? 8
Cost of asset? 1000
Month and year purchased? 09/83
Salvage value? 100

Depreciation from

09/01/83 to 12/31/83 is	\$66.67
01/01/84 to 12/31/84 is	\$191.67
01/01/85 to 12/31/85 is	\$166.67
01/01/86 to 12/31/86 is	\$141.67
01/01/87 to 12/31/87 is	\$116.67
01/01/88 to 12/31/88 is	\$91.67

Press <ENTER> to continue, M for Menu

01/01/89 to 12/31/89 is	\$66.67
01/01/90 to 12/31/90 is	\$41.67
01/01/91 to 09/01/91 is	\$16.67

Remaining salvage value is \$100.00

Press <ENTER> to continue, M for Menu

Line Number and Variable Cross-reference

Ref	Line Numbers			
+-----+				
:02600:	2725			
:02625:	2630	2635		
:02730:	2665	2710		
:10000:	2710	2725		
:C	: 2620	2655	2690	2720
:	Cost of asset			
:D	: 2655	2670		
:	Depreciation Expense			
:D\$: 2625	2630	2635	2640
:	Purchase date			
:D1	: 2675	2685	2690	
:	Yearly deprecitaion			
:F	: 2670	2675	2705	
:	Yearly depreciation			
:L	: 2615	2650	2660	2670 2690
:	Useful lifespan			
:MM	: 2640	2660	2675	2700 2705
:	Month purchased			
:P	: 2675	2705		
:	Miscellaneous variable			
:S	: 2645	2655		
:	Salvage value			
:SY	: 2650	2655		
:	Sum of years digits			
:U\$: 2610	2685	2720	
:	PRINTUSING format field			
:W	: 2660	2680	2700	
:	Last month of depreciation			
:X	: 2665	2680	2700	2710
:	Year counter			
:Y	: 2660	2665	2700	
:	Year purchased			
:YY	: 2640	2680		
:	Year purchased			
:Z	: 2660	2680	2695	
:	Current printed month			
+-----+				

Sum-of-Year's-Digits Depreciation Program

```

2600 CLS:PRINT"DEPRECIATION, SUM-OF-YEAR
'S DIGITS (SYD)":REM B2
2605 ' EXPENSE= (COST-SALVAGE)*((N+1-YR#
)/SUM OF YRS NOS.)
2610 U$="$$#,#####.##"
2615 INPUT"Life of asset in years";L
2620 INPUT"Cost of asset";C
2625 INPUT"Month and year purchased (MM/
YY)";D$
2630 IF LEN(D$)<>5 THEN 2625
2635 IFVAL(LEFT$(D$,2))<1ORVAL(LEFT$(D$,
2))>12THEN2625
2640 MM=VAL(LEFT$(D$,2)):YY=VAL(RIGHT$(D
$,2))
2645 INPUT"Salvage Value";S
2650 SY=(L*(L+1))/2
2655 D=(C-S)/SY
2660 Y=L:Z=MM:W=12
2665 GOSUB2730:FOR X=0 TO Y
2670 F=L*D
2675 D1=F*((12-MM+1)/12)+P
2680 PRINTRIGHT$(STR$(Z),2);"/01/";RIGHT
$(STR$(YY+X),2);" to ";RIGHT$(STR$(W),2)
;"/31/";RIGHT$(STR$(YY+X),2);" is: ";
2685 PRINTUSINGU$;D1
2690 C=C-D1:L=L-1
2695 Z=1
2700 IF X=Y-1 THEN W=MM
2705 P=F*(1-((12-MM+1)/12))
2710 IF(X+1)/6=INT((X+1)/6)THENGOSUB1000
0:GOSUB2730
2715 NEXT
2720 PRINT"Remaining Salvage value is";:
PRINTUSINGU$;C
2725 GOSUB10000:RUN2600
2730 CLS:PRINT"Depreciation from":RETURN
10000 PRINT@281,"Press <ENTER> to contin
ue, M for MENU";
10005 A$=INKEY$:IFA$="M"ORA$="m"THENMAXF
ILES=0:CLEARO:MENUELSEIFA$<>CHR$(13)THEN
10005ELSERETURN

```


DECLINING-BALANCE DEPRECIATION

This method of depreciation uses a fixed percentage of the current value of the asset as the amount to be deducted for the current year. As the value of the asset declines, so does the depreciation, but the percentage remains a constant.

The depreciation for a year is calculated by multiplying the value of the asset by a fixed percentage, usually 150 or 200 percent of the straight-line depreciation rate. The amount is then subtracted from the current value to give next year's current value.

One important fact needs to be pointed out: the depreciation doesn't stop at the end of the life of the asset. Instead, it is considered finished when the salvage value of the asset is reached. If these two events occur in the same year, or the item is fully depreciated before the end of the asset's lifespan, you're okay. If they don't, and the asset's lifespan will be reached before it has been fully depreciated, you should probably switch over to straight-line depreciation at the year when straight-line depreciation is more than declining-balance depreciation.

The next program, Straight-line versus Declining-Balance Depreciation, discusses this crossover in depreciation.

Examples

<i>Program</i>	<i>Your response</i>
Cost of asset?	1000
Enter factor percentage (150%, 200% or other)?	150
Life of asset in years?	8
Salvage value?	100
Date purchased (MM/YY)?	09/83
Depreciation from	
09/01/83 to 12/31/83 is \$62.50	
01/01/84 to 12/31/84 is \$175.78	
01/01/85 to 12/31/85 is \$142.84	
01/01/86 to 12/31/86 is \$116.04	
01/01/87 to 12/31/87 is \$94.29	
Press <ENTER> to continue, M for Menu	

01/01/88 to 12/31/88 is \$76.61
01/01/89 to 12/31/89 is \$62.24
01/01/90 to 09/31/90 is \$169.70
Remaining salvage value is \$100.00

Press <ENTER> to continue, M for Menu

Cost of asset?	25000
Enter factor percentage (150%, 200% or other)?	200
Life of asset in years?	9
Salvage value?	500
Date purchased (MM/YY)?	02/80

Depreciation from

02/01/80 to 12/31/80 is \$5,092.59
01/01/81 to 12/31/81 is \$4,423.87
01/01/82 to 12/31/82 is \$3,440.79
01/01/83 to 12/31/83 is \$2,676.17
01/01/84 to 12/31/84 is \$2,081.46

Press <ENTER> to continue, M for Menu

01/01/85 to 12/31/85 is \$1,618.92
01/01/86 to 12/31/86 is \$1,259.16
01/01/87 to 12/31/87 is \$979.34
01/01/88 to 02/31/88 is \$2,927.70
Remaining salvage value is \$500.00

Press <ENTER> to continue, M for Menu

Line Number and Variable Cross-reference

Ref	Line Numbers				
+-----+					
:02750:	2880				
:02780:	2780	2790			
:02885:	2810	2850			
:10000:	2850	2880			
:C	: 2760	2795			
:	Asset cost				
:D\$: 2780	2785			
:	Purchase date				
:D1	: 2815	2830	2835		
:	Yearly deprecitaion				
:DL	: 2855	2870			
:	Last year's depreciation				
:F	: 2765	2795			
:	Depreciation factor				
:L	: 2770	2795	2805	2810	
:	Useful lifespan				
:MM	: 2785	2790	2800	2805	2865
:	Purchase month				
:S	: 2775	2875			
:	Salvage value				
:U\$: 2755	2830	2870	2875	
:	PRINTUSING format field				
:W	: 2805	2825			
:	End month of depreciation				
:X	: 2810	2820	2825	2850	2860
:	2865				
:	Year counter				
:X0	: 2795	2815			
:	Declining asset cost				
:X1	: 2800	2815			
:	Month factor				
:X2	: 2800	2845			
:	Reciprocal of X1				
:X4	: 2795	2815			
:	Depreciation factor per year				
:X5	: 2795	2835	2855		
:	Declining asset cost				
:X6	: 2815	2845			


```

:      Miscellaneous variable
:X7   : 2815   2845
:      Depreciation value
:Y    : 2805
:      Remaining life
:YY   : 2785   2790   2820   2825   2860
:      2865
:      Year purchased
:Z    : 2805   2820   2840   2860
:      Start month of depreciation
+-----+

```


Declining-Balance Depreciation Program

```

2750 CLS:PRINT"DEPRECIATION - DECLINING
BALANCE METHOD":REM B3
2755 US="$ $#,#####.##"
2760 INPUT"Cost of asset";C
2765 INPUT"Enter factor percentage (150%
, 200%, or other)";F
2770 INPUT"Life of asset";L
2775 INPUT"Salvage value";S
2780 INPUT"Date purchased (MM/YY)";D$:IF
LEN(D$)<>5 THEN 2780
2785 MM=VAL(LEFT$(D$,2)):YY=VAL(RIGHT$(D
$,2))
2790 IFMM<10RMM>12ORYY<0THEN2780
2795 X0=C:X5=C:X4=(F/100)/L
2800 X1=((12-MM+1)/12):X2=1-X1
2805 Y=L:Z=MM:W=12
2810 GOSUB2885:FORX=1TOL-1
2815 X7=X0*X4:X0=X0-X7:D1=X7*X1+X6
2820 PRINTRIGHT$(STR$(Z),2);"/01/";RIGHT
$(STR$(YY+X-1),2);" to ";
2825 PRINTRIGHT$(STR$(W),2);"/31/";RIGHT
$(STR$(YY+X-1),2);" is: ";
2830 PRINTUSINGU$;D1
2835 X5=X5-D1
2840 Z=1
2845 X6=X7*X2
2850 IFX=5THENGOSUB10000:GOSUB2885
2855 NEXT:DL=X5-S
2860 PRINTRIGHT$(STR$(Z),2);"/01/";RIGHT
$(STR$(YY+X-1),2);" to ";
2865 PRINTRIGHT$(STR$(MM),2);"/31/";RIGH
T$(STR$(YY+X-1),2);" is: ";
2870 PRINTUSINGU$;DL
2875 PRINT"Remaining salvage value is";:
PRINTUSINGU$;S
2880 GOSUB10000:RUN2750
2885 CLS:PRINT"Depreciation from":RETURN
10000 PRINT@281,"Press <ENTER> to contin
ue, M for MENU";
10005 A$=INKEY$:IFA$="M"ORA$="m"THENMAXF
ILES=0:CLEARO:MENUELSEIFA$<>CHR$(13)THEN
10005ELSERETURN

```


STRAIGHT-LINE VERSUS DECLINING-BALANCE DEPRECIATION

The previous program mentioned a flaw with using the declining-balance depreciation method: it's possible to reach the end of the asset's lifespan before the asset has been fully depreciated.

This program takes the declining-balance factor and lifespan, salvage value, and cost of the asset and tells you in which year of the depreciation schedule you should switch over to straight-line depreciation.

Examples

<i>Program</i>	<i>Your response</i>
Life of asset?	8
Salvage value?	100
Cost of asset?	1000
Factor declining percent?	150
Cross over to straight-line depreciation for year #5.	
Press <ENTER> to continue, M for Menu	
Life of asset?	9
Salvage value?	500
Cost of asset?	25000
Factor declining percent?	200
Cross over to straight-line depreciation for year #6.	
Press <ENTER> to continue, M for Menu	

Line Number and Variable Cross-reference

Ref	Line Numbers
+-----+	
:02900:	2975
:10000:	2975
:F	: 2925 2930
:	Depreciation factor
:I	: 2940 2945
:	Calculation variable
:L	: 2910 2935
:	Asset lifespan
:N	: 2940 2945
:	Periods in future value
:P	: 2940 2945 2950
:	Principal calculation
:SL	: 2955 2960
:	Straight-line expense
:T	: 2945
:	Loop counter
:X	: 2935 2965 2970
:	Lifespan loop counter
:X0	: 2910 2930 2955 2965
:	Lifespan of asset
:X1	: 2915 2955
:	Salvage value
:X2	: 2950 2960
:	Miscellaneous variable
:X4	: 2930 2940 2950
:	Miscellaneous variable
:X6	: 2920 2940
:	Asset cost
:X9	: 2950 2955
:	Stores value of P
:Y	: 2935 2940 2955
:	Current life value, less one
+-----+	

Straight-line versus Declining-Balance Depreciation Program

```
2900 CLS:PRINT"  Straight Line or Declin
ing Balance?":REM B4
2905 PRINTTAB(10)"When to cross over."
2910 INPUT"Life of asset";L:X0=L
2915 INPUT"Salvage value";X1
2920 INPUT"Cost of asset";X6
2925 INPUT"Factor declining percent";F
2930 X4=-1*((F/X0)/100)
2935 FOR X=1 TO L:Y=X-1
2940 N=Y:P=X6:I=X4
2945 FOR T=1TON:P=P+(P*(I)):NEXT T
2950 X2=-1*(P*X4):X9=P
2955 SL=(X9-X1)/(X0-Y)
2960 IF X2-SL>0 THEN NEXT
2965 IF X>X0 THEN PRINT"Do not cross ove
r to straight line de- preciation."
2970 PRINT"Cross over to straight line d
epreciationfor year #";X;". "
2975 GOSUB10000:RUN2900
10000 PRINT@281,"Press <ENTER> to contin
ue, M for MENU";
10005 A$=INKEY$:IF A$="M"OR A$="m" THEN MAXF
ILES=0:CLEARO:MENUELSEIF A$<>CHR$(13) THEN
10005 ELSE RETURN
```


AMORTIZATION

Amortization is the process of spreading the cost of an item or collection of items over a period of time, with part of each period's cost being a payment on the principal amount and part being interest on the principal. This program generates a schedule of all the payments of an amortized loan.

With this program, you have a choice of having the payment schedule listed to the display, memory, or a printer. For printing in RAM, you'll need to supply a filename for the data file.

Examples

Program

*Your
response*

Print schedule to:

1—Screen

2—Lineprinter

3—File

?

Filename?

Original debt?

Interest rate %?

Number of payments per year?

Total number of payments?

Amortization Schedule

Working

Press <ENTER> to continue, M for Menu

#!	PAYMENT!	INTEREST!	PRINC.!	BALANCE
1!	3314.26!	900.00!	2414.26!	42585.74
2!	3314.26!	851.71!	2462.55!	40123.19
3!	3314.26!	802.46!	2511.80!	37611.39
4!	3314.26!	752.23!	2562.03!	35049.36
5!	3314.26!	700.99!	2613.27!	32436.09
6!	3314.26!	648.72!	2665.54!	29770.55
7!	3314.26!	595.41!	2718.85!	27051.70
8!	3314.26!	541.03!	2773.23!	24278.47
9!	3314.26!	485.57!	2828.69!	21449.78
10!	3314.26!	429.00!	2885.26!	18564.52
11!	3314.26!	371.29!	2942.97!	15621.55

12!	3314.26!	312.43!	3001.83!	12619.72
13!	3314.26!	252.39!	3061.87!	9557.85
14!	3314.26!	191.16!	3123.10!	6434.75
15!	3314.26!	128.70!	3185.56!	3249.19
16!	3314.26!	64.98!	3249.28!	-0.09

Above is an example of how the file TEST.DO looks.

Print schedule to:

1—Screen

2—Lineprinter

3—File

?

1

Original debt?

50000

Interest rate %?

15

Number of payments per year?

12

Total number of payments?

24

Amortization Schedule

Working

Press <ENTER> to continue, M for Menu

#!	PAYMENT!	INTEREST!	PRINC.!	BALANCE
1!	2424.33!	625.00!	1799.33!	48200.67
2!	2424.33!	602.51!	1821.82!	46378.85
3!	2424.33!	579.74!	1844.59!	44534.26
4!	2424.33!	556.68!	1867.65!	42666.61

Press <ENTER> to continue, M for Menu

#!	PAYMENT!	INTEREST!	PRINC.!	BALANCE
5!	2424.33!	533.33!	1891.00!	40775.61
6!	2424.33!	509.70!	1914.63!	38860.98
7!	2424.33!	485.76!	1938.57!	36922.41
8!	2424.33!	461.53!	1962.80!	34959.61
9!	2424.33!	437.00!	1987.33!	32972.28

Press <ENTER> to continue, M for Menu

#!	PAYMENT!	INTEREST!	PRINC.!	BALANCE
10!	2424.33!	412.15!	2012.18!	30960.10
11!	2424.33!	387.00!	2037.33!	28922.77
12!	2424.33!	361.53!	2062.80!	26859.97
13!	2424.33!	335.75!	2088.58!	24771.39
14!	2424.33!	309.64!	2114.69!	22656.70

Press <ENTER> to continue, M for Menu

#!	PAYMENT!	INTEREST!	PRINC.!	BALANCE
15!	2424.33!	283.21!	2141.12!	20515.58
16!	2424.33!	256.44!	2167.89!	18347.69
17!	2424.33!	229.35!	2194.98!	16152.71
18!	2424.33!	201.91!	2222.42!	13930.29
19!	2424.33!	174.13!	2250.20!	11680.09

Press <ENTER> to continue, M for Menu

#!	PAYMENT!	INTEREST!	PRINC.!	BALANCE
20!	2424.33!	146.00!	2278.33!	9401.76
21!	2424.33!	117.52!	2306.81!	7094.95
22!	2424.33!	88.69!	2335.64!	4759.31
23!	2424.33!	59.49!	2364.84!	2394.47
24!	2424.33!	29.93!	2394.40!	0.07

Press <ENTER> to continue, M for Menu

Line Number and Variable Cross-reference

Ref	Line Numbers			
+-----+				
:04600:	4750			
:04615:	4640			
:04640:	4630			
:04645:	4620	4625	4635	
:04750:	4755			
:04755:	4715	4745		
:04765:	4700			
:10000:	4750	4760		
:C	: 4665	4675		
:	Payments per year			
:CO	: 4645	4705	4720	4725
:	Calculation variable			
:D	: 4650	4705	4720	4725 4730
:	Original date			
:EN	: 4745	4755		
:	End of job marker			
:I	: 4710	4715	4730	4740
:	Payment number			
:I1	: 4720	4725	4730	
:	Calculation variable			
:L\$: 4685	4690	4695	4730 4765
:	Print character			
:M\$: 4685	4690	4695	4765
:	Print character			
:N	: 4670	4705	4710	
:	number of payments			
:NF	: 4615	4620	4625	4630 4685
:	4690	4695	4700	4755 4765
:	output to (device)			
:P	: 4705	4725	4730	
:	Calculation variable			
:PU\$: 4605	4730		
:	PRINTUSING format field			
:R	: 4655	4660	4675	
:	Interest rate			
:R1	: 4675	4705	4720	
:	Rate per payment			
+-----+				

Amortization Program

```

4600 MAXFILES=1: CLEAR 900: REM AMORT
4605 PU$="##\\####.##\\####.##\\####.##\\
\\####.##": ' \ is <GRPH> - (minus)
4610 CLS: PRINT TAB(7) "Amortization Schedu
le"
4615 PRINT "Print schedule to: ": PRINT " 1-
Screen": PRINT " 2-Lineprinter": PRINT " 3-F
ile": INPUT NF: CLS
4620 IF NF=1 THEN OPEN "LCD: " FOR OUTPUT AS 1: GO
TO 4645
4625 IF NF=2 THEN OPEN "LPT: " FOR OUTPUT AS 1: GO
TO 4645
4630 IF NF<>3 THEN 4640 ELSE INPUT "Filename";
FN$: IF LEN(FN$)>6 THEN FN$=LEFT$(FN$,6)
4635 FN$=FN$+".DO": OPEN FN$ FOR OUTPUT AS 1: G
O TO 4645
4640 GOT 4615
4645 CO=.5
4650 INPUT "Original debt amount"; D
4655 INPUT "Interest rate %"; R
4660 R=R/100
4665 INPUT "Number of payments per year";
C
4670 INPUT "Total number of payments"; N
4675 R1=R/C
4680 CLS
4685 IF NF=1 OR NF=3 THEN L$=CHR$(245): M$=CHR
$(241)
4690 IF NF=2 THEN L$="!": M$="-"
4695 IF NF=2 OR NF=3 THEN: PRINT "Amortization
Schedule": PRINT @136, "Working": PRINT #1, "
# "; L$; " PAYMENT"; L$; " INTEREST"; L$; " PRIN
C. "; L$; " BALANCE": PRINT #1, STRING$(39, M$
)
4700 IF NF=1 THEN GOSUB 4765
4705 P=INT(CO+D*(R1/(1-(1+R1)^(-N)))*100
)/100
4710 FOR I=1 TO N
4715 IF INT(I/5)=I/5 THEN GOSUB 4755
4720 I1=INT(CO+D*R1*100)/100
4725 D=INT(CO+(D-(P-I1))*100)/100
4730 PRINT #1, USING PU$; I, L$, P, L$, I1, L$, P-
I1, L$, D
4735 PRINT #1, " ";

```



```
4740 NEXT I
4745 EN=1:GOSUB4755
4750 GOSUB10000:RUN4600
4755 IFNF=2THENRETURNELSEIFNF=3THENCLOSE
:OPENFN$FORAPPENDAS1:RETURNELSE IFEN=1TH
EN4750
4760 GOSUB10000
4765 CLS:PRINT#1,"# ";L$;" PAYMENT"L$;"I
NTEREST";L$;" PRINC. ";L$;" BALANCE":PRI
NTSTRING$(39,M$):IFNF=2THENRETURNELSEPRI
NT@42,CHR$(250);:PRINT@51,CHR$(250);:PRI
NT@60,CHR$(250);:PRINT@69,CHR$(250):RETU
RN
10000 PRINT@281,"Press <ENTER> to contin
ue, M for Menu";
10005 A$=INKEY$:IFA$="M"ORA$="m"THENMAXF
ILES=0:CLEARO:MENUELSEIFA$<>CHR$(13)THEN
10005ELSERETURN
```


AVERAGE YIELD OF AN INVESTMENT

When dealing with stock investments, you are usually presented with figures attesting as to yield rate. Since there are many ways to figure the yield rates of stocks, it is easy to become confused as to which stocks are the best investments. To reduce the confusion and simplify the comparisons, it's best to convert the figures to the average annual rate of return, expressed as a percentage.

This program takes the original stock price (the price you paid for it), the current (or projected) stock price, the amount of the dividends paid by the stocks, if any, and the number of periods the stock is held (or invested). From these the program calculates the interest rate, per period, that you earned on the stock (your return on investment).

The program ignores the effect any sales fees or commissions might have on the yield rate.

Examples

<i>Program</i>	<i>Your response</i>
Enter original stock price?	100
Enter current stock price?	120
Enter amount of dividends?	12
Number of periods invested?	5
Interest rate is 5.700% per period. Press <ENTER> to continue, M for Menu	
Enter original stock price?	25
Enter current stock price?	32
Enter amount of dividends?	2
Number of periods invested?	2
Interest rate is 16.610% per period. Press <ENTER> to continue, M for Menu	

Enter original stock price? 25
Enter current stock price? 30
Enter amount of dividends? ENTER
Number of periods invested? 3
Interest rate is 6.260% per period.
Press <ENTER> to continue, M for Menu

Line Number and Variable Cross-reference

Ref	Line Numbers
+-----+	
:03150:	3230
:03185:	3205
:03210:	3200
:03230:	3220
:03235:	3195
:10000:	3230
:10005:	10005
:CP :	3160 3175 3200
:	Current stock price
:DV :	3165 3175
:	Dividends
:I :	3175 3185 3210 3220 3225
:	3240
:	Trial interest rate
:IC :	3175 3185 3210 3215
:	Trial rate increment
:IP :	3215
:	Miscellaneous variable
:OP :	3155 3190
:	Original price
:P# :	3190 3200 3240
:	Trial principal
:PE :	3170 3235
:	Periods invested
:T :	3180 3210
:	Decimal places in answer
:X :	3235 3245
:	Loop counter
+-----+	

Average Yield of an Investment Program

```
3150 CLS:PRINT"      AVERAGE YIELD OF AN I
NVESTMENT":REM D1
3155 INPUT"Enter original stock price";O
P
3160 INPUT"Enter current stock price";CP
3165 INPUT"Enter amount of dividends pai
d";DV
3170 INPUT"Number of periods invested";P
E
3175 I=0:IC=1:CP=CP+DV
3180 FOR T=1TO3
3185 I=I+IC
3190 P#=OP
3195 GOSUB 3235
3200 IFP#>=CP THEN 3210
3205 GOTO 3185
3210 I=I-IC:IC=IC/10:NEXT T
3215 IC=IC*10:IP=IP+IC
3220 IF I<0 THEN PRINT"DATA ERROR. PLEAS
E CHECK YOUR VALUES.":GOTO3230
3225 PRINTUSING"Interest rate is #,###.#
###% per period.";I
3230 GOSUB10000:RUN3150
3235 FOR X=1 TO PE
3240 P#=P#+P#*(I/100)
3245 NEXT X:RETURN
10000 PRINT@281,"Press <ENTER> to contin
ue, M for MENU";
10005 A$=INKEY$:IFA$="M"ORA$="m"THENMAXF
ILES=0:CLEARO:MENUELSEIFA$<>CHR$(13)THEN
10005ELSERETURN
```


WEIGHTED AVERAGE NUMBER OF STOCK SHARES OUTSTANDING

In order for you to compute your business's earnings per share, you must first calculate the total number of shares outstanding. If the total changed during the year, then a weighted average calculation, by days outstanding, needs to be made. This program makes the calculation, ignoring dividends and splits.

First, enter the date for the first day of the period to be considered, in MM/DD/YY format, where MM is the month, DD is the day, and YY is the last two digits of the year. Next, enter the number of shares owned as of that date.

Now enter any dates on which you either purchased or sold stock, followed by the number of shares manipulated. If you purchased stocks, use a positive number; if you sold your stock, use a negative number (precede the number with a minus sign).

Keep doing this until you've entered all the stock transactions. If you haven't any stock transactions or you've already typed them all in, just press <ENTER> in answer to the transaction date prompt. Now type in the last day of the period under consideration.

From this information the program will figure the weighted average number of shares outstanding during the specified period. Your total earnings may then be divided by this number to compute earnings per share.

Note that this program uses a subroutine. This program block, named Days between Dates and printed in chapter one, calculates the number of days between two dates. For more information, see that program's listing.

Examples

<i>Program</i>	<i>Your response</i>
Enter date for first day of period (MM/DD/YY)?	01/01/80
Number of shares outstanding at first period?	1000
Transaction date (press ENTER if done) (MM/DD/YY)?	06/30/80

Enter number of shares purchased?	10000
Transaction date (press ENTER if done)	
(MM/DD/YY)?	07/11/80
Enter number of shares purchased?	2000
Transaction date (press ENTER if done)	
(MM/DD/YY)?	12/02/80
Enter number of shares purchased?	-1000
Transaction date (press ENTER if done)	
(MM/DD/YY)?	01/10/81
Enter number of shares purchased?	100
Transaction date (press ENTER if done)	
(MM/DD/YY)?	ENTER
Enter last day of period	
(MM/DD/YY)?	01/30/81
The weighted average number of shares	
outstanding during the year is:	
7,732.49	
Press <ENTER> to continue, M for Menu	
Enter date for first day of period	
(MM/DD/YY)?	05/01/81
Number of shares outstanding at first	
period?	12345
Transaction date (press ENTER if done)	
(MM/DD/YY)?	12/12/81
Enter number of shares purchased?	2645
Transaction date (press ENTER if done)	
(MM/DD/YY)?	02/02/81
Enter number of shares purchased?	-5000
Transaction date (press ENTER if done)	
(MM/DD/YY)?	ENTER
Enter last day of period	
(MM/DD/YY)?	03/31/82
The weighted average number of shares	
outstanding during the year is:	
12,354.90	
Press <ENTER> to continue, M for Menu	

Line Number and Variable Cross-reference

Ref	Line Numbers				
+-----+					
:03000:	3145				
:03025:	3045	3055	3060		
:03030:	3035	3085			
:03075:	3095	3105	3110	3130	
:03125:	3120				
:03145:	9115				
:09115:	3115				
:09160:	9130	9170			
:09175:	9135	9160	9185		
:09190:	9120	9175			
:09195:	9190				
:10000:	3145				
:D\$: 3030	3035	3040	3050	3060
:	3075	3080	3085	3090	3100
:	3110				
:	Date entered				
:D1	: 3100	3105	3115	9115	9190
:	Day of each transaction				
:DB	: 3115	9125	9145	9165	9180
:	9190				
:	Days between dates				
:DD	: 3050	3055	3115	9115	9125
:	9130	9155	9190		
:	Day of first transaction				
:DD\$: 3050	3100			
:	Month and day				
:E	: 3080	3120	3130		
:	First entry pointer				
:M(: 3020	3025	9125	9145	9180
:	Days in each month				
:M1	: 3090	3095	3115	9115	9120
:	9175				
:	Month of each transaction				
:MM	: 3040	3045	3115	9115	9120
:	9125	9130	9140	9145	9155
:	9175	9180	9185		
:	Month of first transaction				
:P	: 3120				
:	Shares purchased				
:S	: 3070	3120			


```

:      Shares outstanding
:WA   : 3130   3135   3140
:      Weighted average
:X     : 3025   9140   9145   9150
:      Loop counter
:X0    : 3115   3130   3135
:      days outstanding
:X1    : 3065   3070   3125
:      shares outstanding
:X2    : 3125   3130   3135
:      shares outstanding
:X5    : 3115   3125
:      days between dates
:X6    : 3120   3125
:      Total shares outstanding
:X7    : 3125
:      Miscellaneous variable
:Y1    : 3110   3115   9115   9120   9135
:      9160
:      Years of transactions
:YY    : 3060   3115   9115   9120   9125
:      9130   9135   9145   9155   9160
:      9165   9170   9180
:      Year of first transaction
+-----+

```


Weighted Average Number of Stock Shares Outstanding Program

```

3000 CLS:PRINTTAB(6)"WEIGHTED AVG. NUMBE
R OF SHARES":REM C1
3005 ' FOR COMPUTATION OF EARNINGS PER S
HARE
3010 ' BASED ON ACTUAL DAYS HELD
3015 ' STOCK SPLITS AND DIVIDENDS ARE IG
NORED
3020 DIM M(12):DATA 31,28,31,30,31,30,31
,31,30,31,30,31
3025 FOR X=1 TO 12:READM(X):NEXT X
3030 PRINT"Enter date for first day of p
eriod":INPUT"(MM/DD/YY)";D$
3035 IF LEN(D$)<>8 THEN 3030
3040 MM=VAL(LEFT$(D$,2))
3045 IFMM<10RMM>12THEN3025
3050 DD$=LEFT$(D$,5):DD=VAL(RIGHT$(DD$,2
))
3055 IFDD<10RDD>31THEN3025
3060 YY=VAL(RIGHT$(D$,2)):IFY1<0THEN3025
3065 INPUT"Number of shares outstanding
at first period";X1
3070 S=X1
3075 D$="":INPUT"Transaction date (press
ENTER if done) (MM/DD/YY)";D$
3080 IF D$="" THEN PRINT"Enter last day
of period":INPUT"(MM/DD/YY)";D$:E=1
3085 IF LEN(D$)<>8 THEN 3030
3090 M1=VAL(LEFT$(D$,2))
3095 IFM1<10RM1>12THEN3075
3100 DD$=LEFT$(D$,5):D1=VAL(RIGHT$(DD$,2
))
3105 IFD1<10RD1>31THEN3075
3110 Y1=VAL(RIGHT$(D$,2)):IFY1<0THEN3075
3115 GOSUB9115:X5=DB:X0=X0+DB:MM=M1:YY=Y
1:DD=D1
3120 IF E=1 THEN 3125 ELSE INPUT"Enter n
umber of shares purchased";P:S=S+P:X6=S
3125 X7=X1*X5:X2=X2+X7:X1=X6
3130 IF E=0 THEN3075:WA=X2/X0
3135 WA=X2/X0

```



```
3140 PRINT"The weighted average number o
f shares outstanding during the year i
s:":PRINTUSING"#,#####.##";WA
3145 GOSUB10000:RUN3000
9100 ' DAY BETWEEN DATES DBD
9105 ' ***** SUBROUTINE *****
9110 ' M1/D1/Y1 - MM/DD/YY
9115 IF YY>Y1 OR (YY=Y1 AND MM>M1) OR (Y
Y=Y1 AND MM=M1 AND DD>D1) THEN PRINT"Dat
a entry error":GOTO3145
9120 IF Y1=YY AND M1=MM THEN 9190
9125 DB=M(MM)-(DD-1):IF INT(YY/4)=YY/4 A
ND MM=2 THEN DB=DB+1
9130 DD=1:MM=MM+1:IFMM=13 THEN MM=1:YY=Y
Y+1:GOTO9160
9135 IF Y1=YY THEN 9175
9140 FOR X=MM TO 12
9145 DB=DB+M(X):IF INT(YY/4)=YY/4 AND MM
=2 THEN DB=DB+1
9150 NEXT X
9155 MM=1:DD=1:YY=YY+1
9160 IF Y1=YY THEN 9175
9165 DB=DB+365:IF INT(YY/4)=YY/4 THEN DB
=DB+1
9170 YY=YY+1:GOTO9160
9175 IFM1=MM THEN 9190
9180 DB=DB+M(MM):IF INT(YY/4)=YY/4 AND M
M=2 THEN DB=DB+1
9185 MM=MM+1:GOTO9175
9190 IF DD=D1 THEN 9195 ELSE DB=DB+(D1-D
D)
9195 RETURN
10000 PRINT@281,"Press <ENTER> to contin
ue, M for MENU";
10005 A$=INKEY$:IFA$="M"ORAS$="m"THENMAXF
ILES=0:CLEARO:MENUELSEIFA$<>CHR$(13)THEN
10005ELSERETURN
```




4

Time Is Money

It took Albert Einstein years of work and study to develop his theory of the time-space relationship. Jack the Shark realized, after only a few moments in the loan business, that time *is* money. ("If ya wanna borrow my money, ya hafta *pay* fer it . . . daily!")

Money is used as capital to make more money. That earned additional money has many names: interest, profit, opportunity cost, return on investment, internal rate of return, overage, cost of capital, IRS bait, and more. With every transaction made, some provision for the time value of money must be made. These programs allow you to make that provision and calculate the time value of money in every situation.

Each program title in this chapter includes a formula of the sort $A, B, C = X$, where A, B, and C represent values you must supply, and X represents what the program tells you after its calculations. A, B, and C are input, in other words, and X is output. The actual variables used are: FV (future value), PV (present value), PMT (payment), N (number of payments), and I (interest rate). These formulas will let you quickly and accurately find the program that matches your requirements.

Simple questions like How much money will I make if I deposit \$100 a month for ten years? are answered by the simple and compound interest programs. More complex questions dealing with business decisions are addressed by some of the other programs. For example, if your company has a policy of making a 45 percent profit per year on its money and wants to lease a

machine for \$100 a month, what does that machine have to produce by the end of its life to be a worthwhile investment? To calculate this, you would use the annuity program that calculates future value. The formula you would look for in this case would be $PMT, N, I = FV$, where the PMT (payment) is \$100, I (interest rate) is 45 percent, and N (number of periods) is the life in years of the machine. The interest rate would be assumed to be compounding yearly because the company expects its 45 percent profit on an annual basis.

Choose the programs you would like to use, and enter them in the computer. If you like, you may enter them together with the menu program suggested earlier. Keep in mind that there is no need to enter any text preceded by REM or ' unless you would like to modify the programs later.

SIMPLE INTEREST— FUTURE VALUE ($PV, I, N = FV$)

Nothing is free, so when you borrow money from others, they charge you for the privilege. *Interest* is the fee you pay for the use of someone else's money.

There are many different methods of calculating this interest. Some involve simple calculations; others, complex calculations. Almost always, the interest paid is based on four factors:

1. The amount of money borrowed (called the *principal*).
2. The interest rate or fee for using the borrowed money (usually expressed as a percentage of the principal per specific time unit).
3. The length of time the money is to be borrowed, with the most common units being months or years.
4. The method used to calculate the fee or interest.

The last factor is the most important in many cases, since the same numbers used for the first three factors can yield markedly different results in the amount of money that has to be returned to the person it was borrowed from, depending on the method used to calculate the interest.

The first and most obvious method of calculating interest is that used in figuring what is called *simple interest*. You just take the amount borrowed, multiply it by the interest rate (a percentage), and multiply that number by the total number of periods of time the money is borrowed. Thus, borrowing \$1,000 for two years at 10 percent interest per year would require that \$1,200 be returned to the lender.

The time periods for interest calculations are very important. If you are borrowing money and the interest period is one year, then this means that the interest calculation is performed only once a year. Borrowing the money for ten years would mean that the principal is multiplied by the interest rate ten times. On the other hand, if the interest rate is calculated on a monthly basis, then each and every month the principal is multiplied by the interest rate.

The first program in this series is a program for calculating simple interest. The program asks for the principal amount (money to be borrowed), the length of the interest period (*year*, *month*, or *day*—type in one of these three), the interest rate used for each period, and the total number of periods the money is to be borrowed. With this information, the program calculates the total money to be repaid the lender.

Examples

<i>Program</i>	<i>Your response</i>
Principal (dollars)?	1000
Interest period (year, month, day)?	Year
Interest rate (% per year)?	6
Number of years?	2
Total after 2 years is \$1120.00	
Press <ENTER> to continue, M for Menu	
Principal (dollars)?	1000
Interest period (year, month, day)?	Year
Interest rate (% per year)?	12
Number of years?	3
Total after 3 years is \$1360.00	
Press <ENTER> to continue, M for Menu	

Line Number and Variable Cross-reference

Ref	Line Numbers
+-----+	
:00200:	240
:10000:	240
:I	: 215 230
:	Interest rate per time period
:P	: 205 230 235
:	Principal
:P\$: 210 215 220 225 235
:	Interest period
:T	: 225 230 235
:	Number of periods
+-----+	

*Simple Interest—
Future Value
(PV,I,N = FV) Program*

```
200 CLS:PRINTTAB(13)"SIMPLE INTEREST":RE
M A1
205 PRINT:INPUT"Principal (dollars)";P
210 INPUT"Interest period (Year, Month,
Day)";P$
215 PRINT"Interest rate (% per ";P$;")";
:INPUT I
220 P$=P$+"s"
225 PRINT"Number of ";P$;:INPUT T
230 P=P+(T*(P*(I/100)))
235 PRINT"Total after";T;P$;" is ";:PRIN
TUSING"$ $#,#####.##";P
240 GOSUB10000:RUN200
10000 PRINT@281,"Press <ENTER> to contin
ue, M for MENU";
10005 A$=INKEY$:IFA$="M"ORA$="m"THENMAXF
ILES=0:CLEARO:MENUELSEIFA$<>CHR$(13)THEN
10005ELSERETURN
```


COMPOUND INTEREST— FUTURE VALUE ($PV, I, N = FV$)

This program calculates compound interest. Compound interest calculations are performed on the balance of the money, not on just the principal amount. If you're borrowing money, the interest is calculated on the outstanding balance at the end of each interest calculation period.

If you're saving money, the interest earned in one period is added to the principal, so that the next interest calculation is on the new balance. Each period, your money is compounded and adds up to more than the previous month's balance.

You tell this program the principal amount, the interest period, the interest rate, the number of times the interest is compounded per interest period, and the number of interest periods the principal is to be compounded. With these, the program calculates how the balance builds each period. The program displays each period calculated, rather than just displaying the final balance after all the interest periods, so you can see how the money compounds.

Examples

<i>Program</i>	<i>Your response</i>
Principal?	1000
Interest period (year, month, day)?	Year
Interest rate (% per year)?	12
Number of times compounded/year?	1
Number of years?	3
Total after 1 period is \$1,120.00	
Total after 2 periods is \$1,254.40	
Total after 3 periods is \$1,404.93	
Press <ENTER> to continue, M for Menu	

Principal?	1000
Interest period (year, month, day)?	Month
Interest rate (% per month)?	1.5
Number of times compounded/month?	1
Number of months?	3
Total after 1 period is \$1,015.00	
Total after 2 periods is \$1,030.23	
Total after 3 periods is \$1,045.68	
Press <ENTER> to continue, M for Menu	

Line Number and Variable Cross-reference

Ref	Line Numbers				
+-----+					
:00250:	325				
:10000:	315	325			
:C	: 270	275	290		
:	Times compounded per period				
:I	: 265	275	295		
:	Interest rate per time period				
:P	: 255	295	310		
:	Principal				
:P\$: 260	265	270	280	285
:	Interest period				
:T	: 285	290			
:	Number of periods				
:X	: 290	300	305	315	
:	Period counter				
+-----+					

Compound Interest— Future Value (PV,I,N = FV) Program

```
250 CLS:PRINTTAB(10)"COMPOUND INTEREST":  
REM A2  
255 PRINT:INPUT"Principal";P  
260 INPUT"Interest Period (Year, Month,  
Day)";P$  
265 PRINT"Interest rate (% per ";P$;")";  
:INPUT I  
270 PRINT"Number of times compounded/";P  
$;:INPUTC  
275 I=I/C  
280 P$=P$+"s"  
285 PRINT"Number of ";P$;:INPUT T  
290 FOR X=1TOT*C  
295 P=P+P*(I/100)  
300 PRINTUSING"Total after### period";X;  
305 IFX>1THENPRINT"s";  
310 PRINTUSING" is$$$ ,#####.##";P  
315 IF X/7=INT(X/7)THEN GOSUB10000:CLS  
320 NEXT  
325 GOSUB10000:RUN250  
10000 PRINT@281,"Press <ENTER> to contin  
ue, M for MENU";  
10005 A$=INKEY$:IFA$="M"ORAS$="m"THENMAXF  
ILES=0:CLEARO:MENUELSEIFA$<>CHR$(13)THEN  
10005ELSERETURN
```


OTHER COMPOUND INTEREST COMPUTATIONS

The next three programs are interrelated. The first one tells you how long it will take for your investment to reach a predetermined level. The second tells you what your initial investment would have to be to reach a given level from a certain initial investment. The last tells you what the interest rate would have to be to give you the desired return from the initial investment over a set period of time. The three can, of course, be used together or separately.

COMPOUND INTEREST— TIME PERIOD ($PV, I, FV = N$)

Every once in a while, you need to know how long it will take for your money, invested at a certain interest rate, to build up to a certain level. This type of computation comes in handy when you're planning for future expenditures and want to know just how long it will take before you can spend your money on the item you want. Or, for example, to put it more dramatically, how long will it take to double your money?

This program takes the original investment amount, the interest rate per interest period, the number of compounding periods in the interest period, and the target amount of money and then calculates how many interest periods it will take to reach the target amount.

In most cases, the interest period is months or years.

Examples

<i>Program</i>	<i>Your response</i>
Initial investment amount?	1000
Interest rate %?	12
Per (year, month, day)?	Year
Number of times compounded per year?	12
Desired return?	2000
If you invest \$1000.00 at a rate of 12% per year, compounded 12 times per year, you will have \$2006.76 after 70 compounding periods, or 5.83 years.	
Press <ENTER> to continue, M for Menu	

Initial investment amount?	200
Interest rate %?	1
Per (year, month, day)?	Week
Number of times compounded per week?	1
Desired return?	2000

If you invest \$200.00
at a rate of 1% per week,
compounded 1 times per week,
you will have \$2011.82 after 232
compounding periods, or 232 weeks.

Press <ENTER> to continue, M for Menu

Line Number and Variable Cross-reference

Ref	Line Numbers				
+-----+					
:00450:	540				
:00490:	500				
:00505:	490				
:10000:	540				
:B\$: 455	505	525		
:	PRINTUSING format field				
:C	: 475	485	515	535	
:	Times compounded per period				
:FV	: 480	490			
:	Desired future value				
:I	: 465	485	510		
:	Interest rate				
:I\$: 470	475	510	515	520
:	535				
:	Interest Period				
:P	: 485	490	495	525	
:	Increasing principal				
:PI	: 485	495			
:	Interest rate/compounding period				
:PV	: 460	485	505		
:	Present value				
:X	: 495	525	535		
:	Number of times compounded				
+-----+					

Compound Interest— Time Period (PV,I,FV = N) Program

```

450 CLS:PRINT"HOW LONG WILL IT TAKE? TIM
E COMPUTATION":REM A3
455 PRINT:B$="$ $#,#####.##"
460 INPUT"Initial investment amount";PV
465 INPUT"Interest rate %";I
470 INPUT"Per (Year, Month, Day)";I$
475 PRINT"Number of times compounded per
";I$;:INPUTC
480 INPUT"Desired return";FV
485 PI=I/C:PI=PI/100:P=P*PV
490 IF P>=FV THEN 505
495 P=P+(P*PI):X=X+1
500 GOTO 490
505 CLS:PRINT:PRINT"If you invest";USING
B$;PV
510 PRINTTAB(5)"at a rate of";I;"% per "
;I$;". "
515 PRINT"Compounded";C;"times per ";I$;
". "
520 I$=I$+"s"
525 PRINT"You will have";USINGB$;P;:PRIN
T" after";X
530 PRINT"compounding periods, or";
535 PRINTUSING"###.##";X/C;:PRINT" ";I$;
". "
540 GOSUB10000:RUN450
10000 PRINT@281,"Press <ENTER> to contin
ue, M for MENU";
10005 A$=INKEY$:IFA$="M"OR A$="m"THEN MAXF
ILES=0:CLEARO:MENUELSEIFA$<>CHR$(13)THEN
10005ELSERETURN

```


COMPOUND INTEREST— PRESENT VALUE ($FV, I, N = PV$)

This program is similar to the previous program, Compound Interest—Time Period, in that it makes a time computation, only this time you give the program the future value desired, the interest rate, the interest period, the number of compounding periods per interest period, and the total number of compounding periods to be considered. From this information, the program calculates the initial investment amount that you'll need to reach your goal. For instance, how much will you have in ten years?

Examples

<i>Program</i>	<i>Your response</i>
Interest rate %?	12
Per (year, month, day)?	Year
Number of times compounded per year?	12
Amount to be paid in the future?	1000
After how many compounding periods will it be paid?	120
At an interest rate of 12% per year, compounded 12 times per year, the present value of \$1000 to be paid 120 periods hence is: \$302.99	

Press <ENTER> to continue, M for Menu

Interest rate %?	15
Per (year, month, day)?	Year
Number of times compounded per year?	12
Amount to be paid in the future?	1611
After how many compounding periods will it be paid?	48

At an interest rate of 15% per year,
compounded 12 times per year,
the present value of \$1611
to be paid 48 periods hence is:
\$887.43

Press <ENTER> to continue, M for Menu

Line Number and Variable Cross-reference

Ref	Line Numbers
+-----+	
:00550:	635
:10000:	635
:B\$:	555 620 630
:	PRINTUSING format field
:C :	570 585 615
:	Times compounded per period
:FV :	575 590 620
:	Future value
:I :	560 585 610
:	Interest rate per period
:I\$:	565 570 610 615
:	Interest Period
:N :	580 595 625
:	Number of compounding periods
:P :	590 600 630
:	Increasing principal
:PI :	585 600
:	Interest rate/compounding period
:X :	595 605
:	Number of compounding periods
+-----+	

***Compound Interest—
Present Value
(FV,I,N = PV) Program***

```
550 CLS:PRINTTAB(13)"PRESENT VALUE":REM
A4
555 B$="$ $#,#####.##"
560 PRINT:INPUT"Interest rate %";I
565 INPUT"Per (Year, Month, Day)";I$
570 PRINT"Number of times compounded per
";I$;:INPUTC
575 INPUT"Amount to be paid in the futur
e";FV
580 INPUT"After how many compounding per
iods will it be paid";N
585 PI=I/C:PI=PI/100:' PI=INTEREST RATE
PER COMPOUND PERIOD
590 P=FV
595 FOR X=1 TO N
600 P=P/(PI+1)
605 NEXT X:CLS:PRINT
610 PRINT"At an interest rate of";I;"% p
er ";I$;","
615 PRINT"    compounded ";C;"times per "
;I$;","
620 PRINT"    the present value of";USING
B$;FV
625 PRINT"    to be paid";N;"periods henc
e is:"
630 PRINTUSINGB$;P
635 GOSUB10000:RUN550
10000 PRINT@281,"Press <ENTER> to contin
ue, M for MENU";
10005 A$=INKEY$:IF A$="M"OR A$="m" THEN MAXF
ILES=0:CLEARO:MENUELSEIF A$<>CHR$(13) THEN
10005 ELSE RETURN
```


COMPOUND INTEREST— INTEREST RATE (PV,FV,N = I)

This program takes the initial investment amount and the number of years it's invested and determines what the interest rate would have to be to give the desired future value. For investors this would answer such questions as What rate should I shoot for? and What are they paying me?

Examples

<i>Program</i>	<i>Your response</i>
Enter present value?	100
Enter future value?	1000
Enter years invested?	10
Interest rate is 25.89%	
Press <ENTER> to continue, M for Menu	
Enter present value?	1000
Enter future value?	1500
Enter years invested?	5
Interest rate is 8.44%	
Press <ENTER> to continue, M for Menu	

Line Number and Variable Cross-reference

Ref	Line Numbers
+-----+	
:00650:	725
:00680:	700
:00705:	695
:00725:	715
:00730:	690
:10000:	725
:FV	: 660 695
:	Future Value
:I	: 670 680 705 715 720
:	735
:	Interest rate per period
:IC	: 670 680 705 710
:	Miscellaneous Variable
:IP	: 710
:	Interest rate/compounding period
:N	: 665 730
:	Years to calculate
:P#	: 685 695 735
:	Miscellaneous variable
:PV	: 655 685
:	Present value
:T	: 675 705
:	Loop counter
:X	: 730 740
:	Loop counter
+-----+	

Compound Interest— Interest Rate (PV,FV,N = I) Program

```
650 CLS:PRINTTAB(10)"INTEREST RATE":REM
A5
655 INPUT"Enter present value";PV
660 INPUT"Enter future value";FV
665 INPUT"Enter years invested";N
670 I=0:IC=1
675 FOR T=1TO3
680 I=I+IC
685 P#=PV
690 GOSUB 730
695 IFP#>=FV THEN 705
700 GOTO 680
705 I=I-IC:IC=IC/10:NEXT T
710 IC=IC*10:IP=IP+IC
715 IF I<0 THEN PRINT"DATA ERROR. PLEASE
CHECK YOUR VALUES":GOTO725
720 PRINT"Interest rate is:";I;"%"
725 GOSUB10000:RUN650
730 FOR X=1 TO N
735 P#=P#+P#*(I/100)
740 NEXT X
745 RETURN
10000 PRINT@281,"Press <ENTER> to contin
ue, M for MENU";
10005 A$=INKEY$:IF A$="M"OR A$="m" THEN MAXF
ILES=0:CLEARO:MENUELSEIF A$<>CHR$(13) THEN
10005 ELSE RETURN
```


EQUIVALENT INTEREST— COMPARING COMPOUND PERIODS ($IR(c) = IR(c)$)

This program takes the interest rate charged for one interest period and determines what the effective interest rate is after a given number of compounding periods. In other words, if you give it the monthly interest rate (the nominal rate), the program will tell you the annual or semiannual (or any other multiple of the interest period) interest rate (the effective rate of interest for that period of time). The program answers questions such as: If you borrow money at a rate of 5 percent per quarter, compounded monthly, what's the yearly (twelve-period) rate?

To use the program, just give it the interest rate, interest period, number of compounding periods per interest period, and total number of compounding periods for which you want the interest rate calculated. The program takes these figures and calculates the effective interest rate for the number of compounding periods you specified.

Examples

<i>Program</i>	<i>Your response</i>
Interest rate %?	1.5
Per (year, month, day)?	Month
Compounded how many times per month?	1
Figure effective rate for how many compounding periods?	12
Nominal rate is 1.5% per month, compounded 1 times per month.	
Interest rate per compounding period is 1.5%.	
Effective rate after 12 periods is 19.56%.	
Press <ENTER> to continue, M for Menu	

Interest rate %?	12
Per (year, month, day)?	Year
Compounded how many times per year?	1
Figure effective rate for how many compounding periods?	12
Nominal rate is 12% per year, compounded 1 times per year.	
Interest rate per compounding period is 12.00%.	
Effective rate after 12 periods is 289.60%	

Press <ENTER> to continue, M for Menu

Line Number and Variable Cross-reference

Ref	Line Numbers
+-----+	
:00350:	440
:10000:	440
:C :	370 380 415
:	Times compounded per period
:I :	360 380 410
:	Interest rate per time period
:I\$:	365 370 410 415
:	Interest period
:N :	385 395 430
:	Number of compounding periods
:P :	390 395 405 435
:	Effective interest rate
:PI :	380 395 425
:	Interest rate/compounding period
:U\$:	355 425 435
:	PRINTUSING format field
:X :	395 400
:	Loop counter
+-----+	

Equivalent Interest— Comparing Compound Periods ($IR(c) = IR(c)$) Program

```
350 CLS:PRINTTAB(6)"EQUIVALENT INTEREST
RATES":REM A6
355 U$="###.##"
360 PRINT:INPUT"Interest rate %";I
365 INPUT"Per (Year, Month, Day)";I$
370 PRINT"Compounded how many times per
";I$;:INPUTC
375 REM FIGURE INTEREST RATES PER COMPOU
NDING PERIOD
380 PI=I/C:PI=PI/100
385 INPUT"Figure effective rate for how
many com- pounding periods";N
390 P=1
395 FOR X=1 TO N:P=P+(P*PI)
400 NEXT X
405 P=P-1:P=P*100
410 CLS:PRINT"Nominal rate is";I;"% per
";I$
415 PRINTTAB(5)"compounded";C;"times per
";I$
420 PRINT"Interest rate per compounding"
425 PI=PI*100:PRINTTAB(5)"period is"USIN
GU$;PI;:PRINT"%"
430 PRINT"Effective rate after";N;"perio
ds is:"
435 PRINTTAB(5)USINGU$;P;:PRINT"%"
440 GOSUB10000:RUN350
10000 PRINT@281,"Press <ENTER> to contin
ue, M for MENU";
10005 A$=INKEY$:IFA$="M"ORA$="m"THENMAXF
ILES=0:CLEARO:MENUELSEIFA$<>CHR$(13)THEN
10005ELSERETURN
```


ANNUITIES, ORDINARY AND DUE

The next sixteen programs calculate various *annuities*. An *annuity* is a series of equal payments made at regular periods of time, usually weeks or months. Annuities are divided into two classes: *ordinary annuities* and *annuities due*.

Ordinary annuities have payments that are made at the end of each payment period. Loans are usually based on this type of annuity. *Annuities due* have payments that are made at the beginning of each payment period, in anticipation of some type of service or as in an investment program. Leases and savings programs usually fall into this category.

The first four programs calculate present and future values, both ordinary and with continuous compounding.

ORDINARY ANNUITY— FUTURE VALUE (PMT,N,I = FV)

Let's suppose you're making regular deposits into a savings account; how much money will you have after a number of deposits?

This program takes the amount of the payments, their total number, the number per year, the annual interest rate, and the compounding period and calculates the future value.

Examples

<i>Program</i>	<i>Your response</i>
Enter amount of payments?	177.70
Total number of payments?	12
Number of payments per year?	12
Enter annual interest rate %?	12
Compounded how many times a year?	12
Future value of this ordinary annuity is \$2,253.68	
Press <ENTER> to continue, M for Menu	
Enter amount of payments?	63.81
Total number of payments?	18
Number of payments per year?	12
Enter annual interest rate %?	18
Compounded how many times a year?	12
Future value of this ordinary annuity is \$1,307.43	
Press <ENTER> to continue, M for Menu	

Line Number and Variable Cross-reference

Ref	Line Numbers
+-----+	
:01400:	1465
:01465:	1435
:09000:	1440
:10000:	1465
:C :	1430 1435 9000 9005
:	Compounding periods per year
:CP :	9005 9015
:	Compounding periods/payment
:IC :	9000 9015
:	Interest rate/compounding period
:IP :	1450 9020
:	Interest rate per payment
:IR :	1425 9000
:	Annual interest rate
:N2 :	1420 1435 9005
:	Number of payments per year
:N3 :	1415 1445
:	Total number of payments
:P# :	1450 1460
:	Miscellaneous variable
:PQ :	1410 1450
:	Trial payment amount
:QV :	9015 9020
:	Effective interest rate
:X :	1445 1450 9015
:	Loop counter
+-----+	

Ordinary Annuity— Future Value (PMT,N,I = FV) Program

```

1400 CLS:PRINT" FUTURE VALUE OF AN ORDIN
ARY ANNUITY":REM A7
1410 INPUT"Enter amount of payments";PQ
1415 INPUT"Total number of payments";N3
1420 INPUT"Number of payments per year";
N2
1425 INPUT"Enter annual interest %";IR
1430 INPUT"Compounded how many times a y
ear";C
1435 IF N2>C THENPRINT"YOU MAY NOT MAKE
MORE THAN ONE DEPOSIT PER INTEREST COMPO
UNDING PERIOD. PLEASE RE-ENTER":GOTO1465
1440 GOSUB9000:' FIND EFFECTIVE INTEREST
RATE PER PAYMENT
1445 FOR X=1 TO N3
1450 P#=PQ+(P#+(P#*((IP)/100))):NEXTX
1455 PRINT"FUTURE VALUE OF THIS ORDINARY
ANNUITY IS";
1460 PRINTUSING"$$,#####.##";P#
1465 GOSUB10000:RUN1400
9000 IC=(IR/C)/100:'INTEREST RATE PER CO
MPOUNDING PERIOD
9005 CP=C/N2:'COMPOUNDS PER PAYMENT
9010 'FIND EFFECTIVE INTEREST RATE FOR C
2 PERIODS
9015 QV=1:FOR X=1 TO CP:QV=QV+(QV*IC):NE
XT
9020 IP=(QV-1)*100:'IP=INTEREST PER PAYM
ENT
9025 RETURN
10000 PRINT@281,"Press <ENTER> to contin
ue, M for MENU";
10005 A$=INKEY$:IF A$="M"OR A$="m"THENMAXF
ILES=0:CLEARO:MENUELSEIF A$<>CHR$(13)THEN
10005ELSERETURN

```


ORDINARY ANNUITY— FUTURE VALUE WITH CONTINUOUS COMPOUNDING OF INTEREST, PV KNOWN ($PV, N, I = FV$)

Some savings and loan associations and banks are now charging or paying interest using a technique known as the *continuous compounding* of interest. *Continuous compounding* means the compounding of interest throughout each interest period, rather than just at the end of each period.

This means the institution earns, or pays, slightly more than with the other method.

This program takes the annual compounding rate, the number of years to term, and the present value of the payments and calculates the future value of the balance.

Examples

<i>Program</i>	<i>Your response</i>
Enter annual compounding rate?	18
Enter number of years?	2
Enter present value of payments?	1400
Future value is \$2,006.66	
Press <ENTER> to continue, M for Menu	
Enter annual compounding rate?	12
Enter number of years?	5
Enter present value of payments?	1000
Future value is \$1,822.12	
Press <ENTER> to continue, M for Menu	

Line Number and Variable Cross-reference

Ref	Line Numbers
+-----+	
:02150:	2190
:10000:	2190
:FV :	2180 2185
:	Future value
:I :	2160 2175
:	Annual interest rate
:N :	2165 2175
:	Number of years
:P :	2175 2180
:	FV Calculation variable
:PV :	2170 2180
:	Present value
+-----+	

*Ordinary Annuity—
Future Value with
Continuous Compounding
of Interest, PV Known
(PV,N,I = FV) Program*

```
2150 CLS:PRINT"  CONTINUOUS COMPOUNDING
INTEREST RATE":REM A8
2155 PRINTTAB(12)"FUTURE VALUE"
2160 PRINT:INPUT"Enter annual compoundin
g rate";I
2165 INPUT"Enter number of years";N
2170 INPUT"Enter present value of paymen
ts";PV
2175 P=((I/100)*N)
2180 FV=EXP(P)*PV
2185 PRINTUSING"Future value is $$#,###
##.##";FV
2190 GOSUB10000:RUN2150
10000 PRINT@281,"Press <ENTER> to contin
ue, M for MENU";
10005 A$=INKEY$:IFA$="M"ORA$="m"THENMAXF
ILES=0:CLEARO:MENUELSEIFA$<>CHR$(13)THEN
10005ELSERETURN
```


ORDINARY ANNUITY— FUTURE VALUE WITH CONTINUOUS COMPOUNDING OF INTEREST, PMT KNOWN (PMT,N,I = FV)

The previous program used the compounding rate, number of periods, and present value to find the future value. This program uses compounding rate, number of periods, and payment to find the future value of the annuity.

Examples

<i>Program</i>	<i>Your response</i>
Enter periodic compounding rate?	18
Enter number of periods?	3
Amount of payment per period?	12000
Future value is \$47,733.79	
Press <ENTER> to continue, M for Menu	
Enter periodic compounding rate?	12
Enter number of periods?	6
Amount of payment per period?	6000
Future value is \$52,721.66	
Press <ENTER> to continue, M for Menu	

Line Number and Variable Cross-reference

Ref	Line Numbers
+-----+	
:02250:	2290
:10000:	2290
:A :	2275 2280
:	Payment per period
:FV :	2280 2285
:	Future value
:I :	2260 2265 2280
:	Periodic interest rate
:N :	2270 2280
:	Number of periods
:P :	2280
:	FV calculation variable
+-----+	

*Ordinary Annuity—
Future Value with
Continuous Compounding
of Interest, PMT Known
(PMT,N,I = FV) Program*

```
2250 CLS:PRINT" CONTINUOUS COMPOUNDING A
NNUITIES RATE":REM A9
2255 PRINTTAB(12)"FUTURE VALUE"
2260 PRINT:INPUT"Enter periodic compound
ing rate";I
2265 I=I/100
2270 INPUT"Enter number of periods";N
2275 INPUT"Amount of payment per period"
;A
2280 P=I*N:P=EXP(P):FV=A*((P-1)/I)
2285 PRINTUSING"Future value is $$#,####
##.##";FV
2290 GOSUB10000:RUN2250
10000 PRINT@281,"Press <ENTER> to contin
ue, M for MENU";
10005 A$=INKEY$:IFA$="M"ORA$="m"THENMAXF
ILES=0:CLEARO:MENUELSEIFA$<>CHR$(13)THEN
10005ELSERETURN
```


ORDINARY ANNUITY— PRESENT VALUE ($PMT, N, I = PV$)

Given the amount and number of payments in the annuity, the number per year, the annual interest rate, and the compounding period, this program will find the present value of the annuity.

Suppose your company is receiving monthly payments of \$177.70 for one year at 12 percent interest, compounded monthly. What is the present value of the annuity?

Examples

<i>Program</i>	<i>Your response</i>
Enter amount of payments?	177.70
Total number of payments?	12
Number of payments per year?	12
Enter annual interest rate %?	12
Compounded how many times a year?	12
The present value is \$2,000.03	
Press <ENTER> to continue, M for Menu	
Enter amount of payments?	63.81
Total number of payments?	18
Number of payments per year?	12
Enter annual interest rate %?	18
Compounded how many times a year?	12
The present value is \$1,000.07	
Press <ENTER> to continue, M for Menu	

Line Number and Variable Cross-reference

Ref	Line Numbers
+-----+	
:01500:	1565
:01565:	1535
:09000:	1540
:10000:	1565
:C :	1530 1535 9000 9005
:	Compounding periods per year
:CP :	9005 9015
:	Compounding periods/payment
:IC :	9000 9015
:	Interest rate/compounding period
:IP :	1550 9020
:	Interest rate per payment
:IR :	1525 9000
:	Annual interest rate
:N2 :	1520 1535 9005
:	Number of payments per year
:N3 :	1515 1545
:	Total number of payments
:P# :	1550 1560
:	Miscellaneous Variable
:PQ :	1510 1550
:	Trial payment
:QV :	9015 9020
:	Effective interest rate
:X :	1545 1555 9015
:	Loop counter
+-----+	

Ordinary Annuity— Present Value (PMT,N,I = PV) Program

```

1500 CLS:PRINT"PRESENT VALUE OF AN ORDIN
ARY ANNUITY":REM A10
1510 INPUT"Enter amount of payments";PQ
1515 INPUT"Total number of payments";N3
1520 INPUT"Number of payments per year";
N2
1525 INPUT"Enter annual interest rate";I
R
1530 INPUT"Compounded how many times a y
ear";C
1535 IF N2>C THENPRINT"YOU MAY NOT MAKE
MORE THAN ONE DEPOSIT PER INTEREST COMP
OUNDING PERIOD. PLEASE RE-ENTER":GOTO156
5
1540 GOSUB9000:' FIND EFFECTIVE INTEREST
RATE PER PAYMENT
1545 FOR X=N3 TO 1 STEP -1
1550 P#=(PQ+P#)/(1+(IP/100)):'MOVE BACK
ONE PMT
1555 NEXT X
1560 PRINTUSING"The present value is: $$
#,#####.##";P#
1565 GOSUB10000:RUN1500
9000 IC=(IR/C)/100:'INTEREST RATE PER CO
MPOUNDING PERIOD
9005 CP=C/N2:'COMPOUNDS PER PAYMENT
9010 'FIND EFFECTIVE INTEREST RATE FOR C
2 PERIODS
9015 QV=1:FOR X=1 TO CP:QV=QV+(QV*IC):NE
XT
9020 IP=(QV-1)*100:'IP=INTEREST PER PAYM
ENT
9025 RETURN
10000 PRINT@281,"Press <ENTER> to contin
ue, M for MENU";
10005 A$=INKEY$:IFA$="M"ORA$="m"THENMAXF
ILES=0:CLEAR0:MENUELSEIFA$<>CHR$(13)THEN
10005ELSERETURN

```


**ORDINARY ANNUITY—
PRESENT VALUE
WITH CONTINUOUS
COMPOUNDING OF
INTEREST, PMT KNOWN
($PMT, N, I = PV$)**

This program solves for the present value of an annuity using the periodic compounding rate, the number of periods, and the payment per period.

Examples

<i>Program</i>	<i>Your response</i>
Enter periodic compounding rate?	18
Enter number of periods?	4
Amount of payment per period?	4000
Present value is \$11,405.51	
Press <ENTER> to continue, M for Menu	
Enter periodic compounding rate?	12
Enter number of periods?	6
Amount of payment per period?	7500
Present value is \$32,077.98	
Press <ENTER> to continue, M for Menu	

Line Number and Variable Cross-reference

Ref	Line Numbers
+-----+	
:02200:	2245
:10000:	2245
:A	: 2225 2230
:	Payment per period
:I	: 2210 2215 2230
:	Periodic interest rate
:N	: 2220 2230
:	Number of periods
:P	: 2230 2235
:	PV Calculation variable
:PV	: 2235 2240
:	Present value
+-----+	

*Ordinary Annuity—
Present Value
with Continuous
Compounding of
Interest, PMT Known
(PMT,N,I = PV) Program*

```
2200 CLS:PRINT"    CONTINUOUS COMPOUNDING
      ANNUITIES":REM A11
2205 PRINTTAB(12)"PRESENT VALUE"
2210 INPUT"Enter periodic compounding ra
      te";I
2215 I=I/100
2220 INPUT"Enter number of periods";N
2225 INPUT"Amount of payment per period"
      ;A
2230 P=-1*(I*N):P=EXP(P):P=A*((1-P)/I)
2235 PV=P
2240 PRINTUSING"Present value is $$#,###
      ###.##";PV
2245 GOSUB10000:RUN2200
10000 PRINT@281,"Press <ENTER> to contin
      ue, M for MENU";
10005 A$=INKEY$:IFA$="M"ORA$="m"THENMAXF
      ILES=0:CLEARO:MENUELSEIFA$<>CHR$(13)THEN
10005ELSERETURN
```


ORDINARY ANNUITY— PRESENT VALUE WITH CONTINUOUS COMPOUNDING OF INTEREST, FV KNOWN (FV,N,I = PV)

This program computes the present value of the payments from the annual compounding rate, the number of years the balance is on deposit, and the future value of the payments.

Examples

<i>Program</i>	<i>Your response</i>
Enter annual compounding rate?	18
Enter number of years?	2
Enter future value of payments?	2000
Present value is \$1,395.35	
Press <ENTER> to continue, M for Menu	
Enter annual compounding rate?	12
Enter number of years?	2
Enter future value of payments?	2000
Present value is \$1,573.26	
Press <ENTER> to continue, M for Menu	

Line Number and Variable Cross-reference

Ref	Line Numbers
+-----+	
:02100:	2140
:10000:	2140
:FV :	2120 2130
:	Future value
:I :	2110 2125
:	Annual interest rate
:N :	2115 2125
:	Number of years
:P :	2125 2130
:	Calculation variable
:PV :	2130 2135
:	Present value
+-----+	

*Ordinary Annuity—
Present Value
with Continuous
Compounding of Interest,
FV Known
(FV,N,I = PV) Program*

```
2100 CLS:PRINT"  CONTINUOUS COMPOUNDING
INTEREST RATE":REM A12
2105 PRINTTAB(12)"PRESENT VALUE"
2110 PRINT:INPUT"Enter annual compoundin
g rate";I
2115 INPUT"Enter number of years";N
2120 INPUT"Enter future value of payment
s";FV
2125 P=((I/100)*N*-1)
2130 PV=EXP(P)*FV
2135 PRINTUSING"Present value is $$#,###
###.##";PV
2140 GOSUB10000:RUN2100
10000 PRINT@281,"Press <ENTER> to contin
ue, M for MENU";
10005 A$=INKEY$:IFA$="M"ORA$="m"THENMAXF
ILES=0:CLEARO:MENUELSEIFA$<>CHR$(13)THEN
10005ELSERETURN
```


ORDINARY ANNUITY— PAYMENT, PV KNOWN ($PV, N, I = PMT$)

If you are purchasing equipment for a business, one of your requirements is to know the cost-to-earnings ratio: how much money the equipment costs per period versus how much money it earns for the company in the same period. If the ratio is too low, the equipment shouldn't be purchased. For example, if you purchase the equipment for \$1,000, and you're using 18 percent per year as your cost of capital, what revenues does the machine need to earn for you to justify its purchase?

This program will determine the cost of the equipment, or the loan payment, when given the starting or present value, the total number of payments, the number of payments per year, the annual interest rate, and the number of compounding periods per year.

Examples

<i>Program</i>	<i>Your response</i>
Enter annuity present value?	1000
Total number of payments?	18
Number of payments per year?	12
Enter annual interest rate %?	18
Compounded how many times a year?	12
Payments would be \$63.81	
Press <ENTER> to continue, M for Menu	
Enter annuity present value?	2000
Total number of payments?	12
Number of payments per year?	12
Enter annual interest rate %?	12
Compounded how many times a year?	12
Payments would be \$177.70	
Press <ENTER> to continue, M for Menu	

Line Number and Variable Cross-reference

Ref	Line Numbers
+-----+	
:01250:	1340
:01300:	1320 1325
:01335:	1315
:01340:	1330
:09000:	1290
:10000:	1340
:C :	1280 9000 9005
:	Compounding periods per year
:CP :	9005 9015
:	Compounding periods/payment
:IC :	9000 9015
:	Interest rate/compounding period
:IP :	1305 9020
:	Interest rate per payment
:IR :	1275 9000
:	Annual interest rate
:N2 :	1270 9005
:	Number of payments per year
:N3 :	1265 1295 1300
:	Total number of payments
:P# :	1305 1315 1320 1325
:	Payment calculation variable
:PP :	1260 1295 1315 1320 1325
:	Present value
:PQ :	1295 1305 1320 1325 1335
:	Trial payments
:QQ :	1285 1320 1325
:	Trial increment
:QV :	9015 9020
:	Effective interest rate
:X :	1300 1310 9015
:	Loop counter
+-----+	

Ordinary Annuity— Payment, PV Known (PV,N,I = PMT) Program

```

1250 CLS:PRINT"  PAYMENT FOR AN ORDINARY
      ANNUITY":REM A13
1255 ' I.E. FIND LOAN PAYMENTS BEGINNING
      1 PERIOD AFTER LOAN
1260 INPUT"Enter annuity present value";
      PP
1265 INPUT"Total number of payments";N3
1270 INPUT"Number of payments per year";
      N2
1275 INPUT"Enter annual interest rate %"
      ;IR
1280 INPUT"Compounded how many times a y
      ear";C
1285 QQ=.1
1290 GOSUB9000:' FIND EFFECTIVE INTEREST
      RATE PER PAYMENT
1295 PQ=PP/N3
1300 FOR X=N3TO 1 STEP -1
1305 P#=(PQ+P#)/(1+(IP/100)):'MOVE BACK
      ONE PMT
1310 NEXT X
1315 IFINT(P#*100)=PP*100 THEN 1335
1320 IF P#<PP THEN PQ=PQ+(PQ*QQ):P#=0:GO
      TO1300
1325 IF P#>PP THEN QQ=QQ*.1:PQ=PQ-(PQ*(Q
      Q*10)):P#=0:GOTO 1300
1330 GOTO1340
1335 PRINT"Payments would be";USING"$$#,
      #####.##";PQ
1340 GOSUB10000:RUN1250
9000 IC=(IR/C)/100:'INTEREST RATE PER CO
      MPOUNDING PERIOD
9005 CP=C/N2:'COMPOUNDS PER PAYMENT
9010 'FIND EFFECTIVE INTEREST RATE FOR C
      2 PERIODS
9015 QV=1:FOR X=1 TO CP:QV=QV+(QV*IC):NE
      XT
9020 IP=(QV-1)*100:'IP=INTEREST PER PAYM
      ENT
9025 RETURN

```

```
10000 PRINT@281,"Press <ENTER> to contin  
ue, M for MENU";  
10005 A$=INKEY$:IF A$="M"OR A$="m"THEN MAXF  
ILES=0:CLEARO:MENUELSE IF A$<>CHR$(13)THEN  
10005 ELSE RETURN
```

ORDINARY ANNUITY— PAYMENT, FV KNOWN (FV,N,I = PMT)

If you know the future value, compounding period, annual interest rate, payment period, and total number of payments, you can use this program to tell you the payment amount. You may also use the program to calculate the necessary deposits for a sinking fund.

Examples

<i>Program</i>	<i>Your response</i>
Enter future value of annuity?	10000
Total number of payments?	15
Number of payments per year?	12
Enter annual interest rate %?	7.4
Compounded how many times a year?	12
Payments would be \$371.19	
Press <ENTER> to continue, M for Menu	
Enter future value of annuity?	1500
Total number of payments?	48
Number of payments per year?	12
Enter annual interest rate %?	17.16
Compounded how many times a year?	12
Payments would be \$21.96	
Press <ENTER> to continue, M for Menu	

Line Number and Variable Cross-reference

Ref	Line Numbers
+-----+	
:01700:	1790
:01750:	1770 1775
:01785:	1765
:09000:	1740
:10000:	1790
:C :	1730 9000 9005
:	Compounding periods per year
:CP :	9005 9015
:	Compounding periods/payment
:FV :	1710 1745 1770 1775
:	Future value
:IC :	9000 9015
:	Interest rate/compounding period
:IP :	1755 9020
:	Interest rate per payment
:IR :	1725 9000
:	Annual interest rate
:N2 :	1720 9005
:	Number of payments per year
:N3 :	1715 1745 1750
:	Total number of payments
:P# :	1755 1765 1770 1775
:	Miscellaneous variable
:PP :	1765
:	Estimated payment
:PQ :	1745 1755 1770 1775 1785
:	Trial payment
:QQ :	1735 1770 1775
:	Trial increment
:QV :	9015 9020
:	Effective interest rate
:X :	1750 1760 9015
:	Loop counter
+-----+	

Ordinary Annuity— Payment, FV Known (FV,N,I = PMT) Program

```

1700 CLS:PRINT"      PAYMENT FOR AN ORDIN
ARY ANNUITY":REM A14
1710 INPUT"Enter future value of annuity
";FV
1715 INPUT"Total number of payments";N3
1720 INPUT"Number of payments per year";
N2
1725 INPUT"Enter annual interest rate %"
;IR
1730 INPUT"Compounded how many times a y
ear";C
1735 QQ=.1
1740 GOSUB9000:' FIND EFFECTIVE INTEREST
RATE PER PAYMENT
1745 PQ=FV/N3
1750 FOR X=1 TO N3
1755 P#=PQ+(P#+(P#*((IP)/100)))
1760 NEXT X
1765 IFINT(P#*100)=FV*100 THEN 1785
1770 IF P#<FV THEN PQ=PQ+(PQ*QQ):P#=0:GO
TO1750
1775 IF P#>FV THEN QQ=QQ*.1:PQ=PQ-(PQ*(Q
Q*10)):P#=0:GOTO 1750
1785 PRINT"Payments would be ";:PRINTUSI
NG"$$#,#####.##";PQ
1790 GOSUB10000:RUN1700
9000 IC=(IR/C)/100:'INTEREST RATE PER CO
MPOUNDING PERIOD
9005 CP=C/N2:'COMPOUNDS PER PAYMENT
9010 'FIND EFFECTIVE INTEREST RATE FOR C
2 PERIODS
9015 QV=1:FOR X=1 TO CP:QV=QV+(QV*IC):NE
XT
9020 IP=(QV-1)*100:'IP=INTEREST PER PAYM
ENT
9025 RETURN
10000 PRINT@281,"Press <ENTER> to contin
ue, M for MENU";
10005 A$=INKEY$:IFA$="M"ORAS$="m"THENMAXF
ILES=0:CLEARO:MENUELSEIFA$<>CHR$(13)THEN
10005ELSERETURN

```


ORDINARY ANNUITY— INTEREST RATE, FV KNOWN ($PMT, N, FV = I$)

If you know the number, period, and amount of the payments and the future value of your annuity, what is the interest rate? Or suppose you are depositing \$100 a month for twelve months, with a goal of having \$1,300 by the end of the annuity. What interest rate do you need?

Examples

<i>Program</i>	<i>Your response</i>
Enter amount of payments?	100
Total number of payments?	12
Number of payments per year?	12
Enter the future value of the annuity?	1300
Interest rate is 1.43% per period, 17.16% per year, compounded 12 times a year. Press <ENTER> to continue, M for Menu	
Enter amount of payments?	50
Total number of payments?	18
Number of payments per year?	12
Enter the future value of the annuity?	1300
Interest rate is 4.14% per period, 49.68% per year, compounded 12 times a year. Press <ENTER> to continue, M for Menu	

Line Number and Variable Cross-reference

Ref	Line Numbers
+-----+	
:01600:	1675
:01640:	1650
:01655:	1640
:01675:	1665
:01680:	1635 1650
:10000:	1675
:FV :	1625 1640
:	Future value
:IC :	1630 1645 1655 1660
:	Interest rate/compounding period
:IP :	1630 1645 1655 1660 1665
:	1670 1685
:	Interest rate per period
:N2 :	1620 1670
:	Number of payments per year
:N3 :	1615 1680
:	Total number of payments
:P# :	1630 1640 1650 1685
:	Miscellaneous variable
:PQ :	1610 1685
:	Trial payment amount
:T :	1630 1655
:	Loop counter
:X :	1680 1690
:	Loop counter
+-----+	

*Ordinary Annuity—
Interest Rate, FV Known
(PMT,N,FV = I) Program*

```

1600 CLS:PRINT" INTEREST RATE OF AN ORDINARY ANNUITY":REM A15
1605 ' FUTURE VALUE GIVEN
1610 INPUT"Enter amount of payments";PQ
1615 INPUT"Total number of payments";N3
1620 INPUT"Number of payments per year";N2
1625 PRINT"Enter the future value of the ":INPUT" annuity";FV
1630 IP=0:IC=1:FORT=1 TO 3:P#=1
1635 GOSUB1680
1640 IFP#>FV THEN 1655
1645 IP=IP+IC
1650 P#=1:GOSUB 1680:GOTO 1640
1655 IP=IP-IC:IC=IC/10:NEXT T
1660 IC=IC*10:IP=IP+IC
1665 IF IP<0 THEN PRINT"DATA ERROR. PLEASE CHECK YOUR VALUES":GOTO1675
1670 CLS:PRINT"Interest rate is";IP;"% per period,":PRINTUSING"###.##";IP*N2;:PRINT"% ";:PRINT"per year, compounded";N2;"times a year."
1675 GOSUB10000:RUN1600
1680 FOR X=1 TO N3
1685 P#=PQ+(P#+(P#*((IP)/100)))
1690 NEXT X
1695 RETURN
10000 PRINT@281,"Press <ENTER> to continue, M for MENU";
10005 A$=INKEY$:IFA$="M"ORA$="m"THENMAXFILES=0:CLEARO:MENUELSEIFA$<>CHR$(13)THEN
10005ELSERETURN

```


ORDINARY ANNUITY— INTEREST RATE, PV KNOWN ($PMT, N, PV = I$)

As an investor, you're probably frequently approached by people wanting your financial support. One of the variables to consider is the rate of return on your money.

This program takes the total number of payments, number of payments per year, amount of the payments, and present value (investment amount) of the payments and calculates the interest rate of your return on your investment.

Examples

<i>Program</i>	<i>Your response</i>
Enter amount of payments?	75
Total number of payments?	16
Number of payments per year?	12
Enter present value of the annuity?	1000
Interest rate is 1.52% per period, 18.24% per year, compounded 12 times a year.	
Press <ENTER> to continue, M for Menu	
Enter amount of payments?	100
Total number of payments?	24
Number of payments per year?	12
Enter present value of the annuity?	2000
Interest rate is 2.24% per period, 26.88% per year, compounded 12 times a year.	
Press <ENTER> to continue, M for Menu	

Line Number and Variable Cross-reference

Ref	Line Numbers
+-----+	
:02000:	2075
:02040:	2050
:02055:	2040
:02075:	2065
:02080:	2035 2050
:10000:	2075
:IC :	2030 2045 2055 2060
:	Interest rate/compounding period
:IP :	2030 2045 2055 2060 2065
:	2070 2085
:	Interest rate per period
:N2 :	2020 2070
:	Number payments per year
:N3 :	2015 2080
:	Total number of payments
:P# :	2030 2040 2050 2085
:	Miscellaneous variable
:PQ :	2010 2085
:	Payment amount
:PV :	2025 2040
:	Present value
:T :	2030 2055
:	Loop counter
:X :	2080 2090
:	Loop counter
+-----+	

Ordinary Annuity— Interest Rate, PV Known (PMT,N,PV = I) Program

```

2000 CLS:PRINT" INTEREST RATE OF AN ORD
INARY ANNUITY":REM A16
2005 PRINTTAB(9)"PRESENT VALUE GIVEN"
2010 INPUT"Enter amount of payments";PQ
2015 INPUT"Total number of payments";N3
2020 INPUT"Number of payments per year";
N2
2025 INPUT"Enter the present value of th
e annuity";PV
2030 IP=0:IC=1:FORT=1 TO 3:P#=1
2035 GOSUB2080
2040 IFP#<PV THEN 2055
2045 IP=IP+IC
2050 P#=1:GOSUB 2080:GOTO 2040
2055 IP=IP-IC:IC=IC/10:NEXT T
2060 IC=IC*10:IP=IP+IC
2065 IF IP<0 THEN PRINT"DATA ERROR. PLEA
SE CHECK YOUR VALUES":GOTO2075
2070 CLS:PRINT"Interest rate is:":PRINTU
SING"###.##% per period,    ###.##% per
year,compounded ### times a year.";IP;IP
*N2;N2
2075 GOSUB10000:RUN2000
2080 FOR X=N3TO 1 STEP -1
2085 P#=(PQ+P#)/(1+(IP/100)):REM MOVE BA
CK ONE PMT
2090 NEXT X
2095 RETURN
10000 PRINT@281,"Press <ENTER> to contin
ue, M for MENU";
10005 A$=INKEY$:IF A$="M"OR A$="m"THEN MAXF
ILES=0:CLEARO:MENUELSEIF A$<>CHR$(13)THEN
10005ELSERETURN

```


ORDINARY ANNUITY— NUMBER OF PAYMENTS (PV,PMT,I = N)

Many times, when borrowing money, you know what you want the monthly payment to be, how much you're going to be borrowing, the interest rate, the payment period, and the compounding period, but you need to find the number of payments and the amount of the final payment. For example, if you were going to buy a \$1,000 piece of equipment and you could afford to pay only \$75 per month, how long would it take to pay it off? This program will provide that information. Payments are assumed to fall at the end of each month.

Examples

<i>Program</i>	<i>Your response</i>
Enter present value of annuity?	1000
Enter amount of payments?	75
Number of payments per year?	12
Enter annual interest rate %?	12
Compounded how many times a year?	12
You will make 15 payments. The last payment will be \$28.70.	
Press <ENTER> to continue, M for Menu	
Enter present value of annuity?	2000
Enter amount of payments?	100
Number of payments per year?	12
Enter annual interest rate %?	12
Compounded how many times a year?	12
You will make 23 payments. The last payment will be \$42.70.	
Press <ENTER> to continue, M for Menu	

Line Number and Variable Cross-reference

Ref	Line Numbers
+-----+	
:01900:	1995
:01950:	1955
:01995:	1935
:09000:	1940
:10000:	1995
:C	: 1930 1935 9000 9005
:	Compounding periods per year
:CP	: 9005 9015
:	Compounding periods/payment
:IC	: 9000 9015
:	Interest rate/compounding period
:IP	: 1950 1965 1985 9020
:	Interest rate per payment
:IR	: 1925 9000
:	Annual interest rate
:N2	: 1920 1935 9005
:	Payments periods per year
:P#	: 1950 1955 1960 1965 1970
:	Payment calculator
:PQ	: 1915 1950 1965
:	Payment amount
:PV	: 1910 1955 1970 1985 1990
:	Present value
:QV	: 9015 9020
:	Effective interest rate
:X	: 1945 1955 1960 1975 1980
:	1990 9015
:	Number of payments
:Z	: 1960 1970 1980 1985
:	Loop counter
+-----+	

Ordinary Annuity— Number of Payments (PV,PMT,I = N) Program

```

1900 CLS:PRINT"  NUMBER OF PAYMENTS, AND
      LAST PAYMENT":REM A17
1905 ' I.E. PAYMENTS MADE AT THE END OF
      EACH MONTH
1910 INPUT"Enter present value of the an
      nuity";PV
1915 INPUT"Enter amount of payments";PQ
1920 INPUT"Number of payments per year";
      N2
1925 INPUT"Enter annual interest rate %"
      ;IR
1930 INPUT"Compounded how many times a y
      ear";C
1935 IF N2>C THENPRINT"YOU MAY NOT MAKE
      MORE THAN ONE DEPOSIT  PER INTEREST COMP
      OUNDING PERIOD. PLEASE RE-ENTER":GOTO199
      5
1940 GOSUB9000:' FIND EFFECTIVE INTEREST
      RATE PER PAYMENT
1945 X=0
1950 P#=(PQ+P#)/(1+(IP/100)):'MOVE BACK
      ONE PMT
1955 IF P#<PV THEN X=X+1:GOTO 1950
1960 P#=0:FOR Z=X TO 1 STEP -1
1965 P#=(PQ+P#)/(1+(IP/100)):'MOVE BACK
      ONE PMT
1970 NEXTZ:PV=PV-P#
1975 X=X+1:'ADD LAST PMT
1980 FOR Z=1 TO X
1985 PV=PV+(PV*(IP/100)):NEXTZ
1990 PRINTUSING"you will make #,### paym
      ents.  The last payment will be $$#,####
      ###.##";X;PV
1995 GOSUB10000:RUN1900
9000 IC=(IR/C)/100:'INTEREST RATE PER CO
      MPOUNDING PERIOD
9005 CP=C/N2:'COMPOUNDS PER PAYMENT
9010 'FIND EFFECTIVE INTEREST RATE FOR C
      2 PERIODS

```



```
9015 QV=1:FOR X=1 TO CP:QV=QV+(QV*IC):NE
XT
9020 IP=(QV-1)*100:'IP=INTEREST PER PAYM
ENT
9025 RETURN
10000 PRINT@281,"Press <ENTER> to contin
ue, M for MENU";
10005 A$=INKEY$:IFA$="M"ORA$="m"THENMAXF
ILES=0:CLEARO:MENUELSEIFA$<>CHR$(13)THEN
10005ELSERETURN
```


ANNUITY DUE— FUTURE VALUE (I,PMT,N = FV)

The most frequent use for this type of program is in determining the future value of regular deposits, made at the beginning of each month, in a savings account. Thus, not only does the bank balance grow by compounding interest each month, it also is increased by the deposit.

To use the program, just give it the dollar amount of the payments, the total number of payments, the number of payments per year, the annual interest rate, and the number of compounding periods in one year. From these the program will calculate and display the growth of the balance, as well as a final total. The balances displayed are the balances as of that deposit; thus, the first deposit represents the opening balance of the account. The final deposit is the total at the end of the month of the last deposit (which was made at the beginning of the month).

If your computation requires that multiple deposits be made in each compounding period, enter the total payment made per compound period. Of course, payments made midperiod will not start earning interest until the start of the next compound period.

Examples

<i>Program</i>	<i>Your response</i>
Enter amount of payments?	100
Total number of payments?	12
Number of payments per year?	12
Enter annual interest rate %?	12
Compounded how many times per year?	12
Payment 1 total on deposit \$100.00	
Payment 2 total on deposit \$201.00	
Payment 3 total on deposit \$303.01	
Payment 4 total on deposit \$406.04	
Payment 5 total on deposit \$510.10	
Payment 6 total on deposit \$615.20	
Payment 7 total on deposit \$721.35	
Press <ENTER> to continue, M for Menu	

Payment 8 total on deposit \$828.57
 Payment 9 total on deposit \$936.85
 Payment 10 total on deposit \$1046.22
 Payment 11 total on deposit \$1156.68
 Payment 12 total on deposit \$1268.25
 Final total \$1280.93

Press <ENTER> to continue, M for Menu

Enter amount of payments?	50
Total number of payments?	12
Number of payments per year?	12
Enter annual interest rate?	18
Compounded how many times per year?	12
Payment 1 total on deposit \$50.00	
Payment 2 total on deposit \$100.75	
Payment 3 total on deposit \$152.26	
Payment 4 total on deposit \$204.55	
Payment 5 total on deposit \$257.61	
Payment 6 total on deposit \$311.48	
Payment 7 total on deposit \$366.15	

Press <ENTER> to continue, M for Menu

Payment 8 total on deposit \$421.64
 Payment 9 total on deposit \$477.97
 Payment 10 total on deposit \$535.14
 Payment 11 total on deposit \$593.16
 Payment 12 total on deposit \$652.06
 Final total \$661.84

Press <ENTER> to continue, M for Menu

Line Number and Variable Cross-reference

Ref	Line Numbers				
+-----+					
:00750:	830				
:09000:	790				
:10000:	785	810	830		
:C	: 780	785	9000	9005	
:	Times compounded in a year				
:CP	: 9005	9015			
:	Compounds per payment				
:IC	: 9000	9015			
:	Interest rate/compounding period				
:IP	: 800	820	9020		
:	Interest per payment				
:IR	: 775	9000			
:	Annual interest rate				
:N2	: 770	785	9005		
:	Number of payments per year				
:N3	: 765	795			
:	Number of payments				
:P#	: 800	805	820	825	
:	Total on deposit				
:PQ	: 760	800			
:	Payment amount				
:QV	: 9015	9020			
:	Effective interest				
:X	: 795	805	810	815	9015
:	Payment number				
+-----+					

Annuity Due— Future Value (I,PMT,N = FV) Program

```

750 CLS:PRINTTAB(3)"FUTURE VALUE OF AN A
    NNUITY DUE":REM A18
755 ' I.E. DEPOSITS MADE INTO A SAVINGS
    ACCOUNT AT THE BEGINNING OF EACH MONTH
760 PRINT:INPUT"Enter amount of payments
    ";PQ
765 INPUT"Total number of payments";N3
770 INPUT"Number of payments per year";N
    2
775 INPUT"Enter annual interest rate %";
    IR
780 INPUT"Compounded how many times per
    year";C
785 IF N2>C THENPRINT"You may not make m
    ore than one deposit per interest compo
    unding period.":GOSUB10000
790 GOSUB9000:' FIND EFFECTIVE INTEREST
    RATE PER PAYMENT
795 FOR X=1 TO N3
800 P#=PQ+(P#+(P#*((IP)/100))):' MOVE FW
    D 1 PMT PERIOD
805 PRINTUSING"Payment ###, total on dep
    osit$$$$###.##";X;P#;
810 IF X/7=INT(X/7)THENGOSUB10000:CLS
815 NEXT X
820 P#=P#+(P#*((IP)/100)):' MOVE FWD LAS
    T PMT PERIOD
825 PRINT"Final total";USING"$$#,#####
    .##";P#
830 GOSUB10000:RUN750
9000 IC=(IR/C)/100:'INTEREST RATE PER CO
    MPOUNDING PERIOD
9005 CP=C/N2:'COMPOUNDS PER PAYMENT
9010 'FIND EFFECTIVE INTEREST RATE FOR C
    2 PERIODS
9015 QV=1:FOR X=1 TO CP:QV=QV+(QV*IC):NE
    XT
9020 IP=(QV-1)*100:'IP=INTEREST PER PAYM
    ENT

```



```
9025 RETURN
10000 PRINT@281,"Press <ENTER> to contin
ue, M for MENU";
10005 A$=INKEY$:IFA$="M"ORA$="m"THENMAXF
ILES=0:CLEARO:MENUELSEIFA$<>CHR$(13)THEN
10005ELSERETURN
```


ANNUITY DUE— PRESENT VALUE (I,PMT,N = PV)

This program calculates the present value (or *starting value*) of an annuity due, given the value of the payment itself. For example, if you sign a ten-year lease for a piece of equipment at \$250 per payment with an interest rate of 18 percent, then what is the present value of the annuity due (what would be an equivalent cash price for the equipment)?

The program asks for the payment amount, the total number of payments, the number of payments made per year, the annual interest rate, and the number of times the balance is compounded per year. Only one payment is allowed per interest compounding period (or no payment at all in that period); that is, if the balance is compounded quarterly, the payments must be quarterly, semi-annual, or annual, and not monthly. This is obvious in that two payments within one compounding period aren't worth more than one sum payment in the same period. From this information the program calculates the present value of the annuity due.

Examples

<i>Program</i>	<i>Your response</i>
Enter amount of payments?	100
Total number of payments?	12
Number of payments per year?	12
Enter annual interest rate %?	12
Compounded how many times a year?	12
Value of annuity due: \$1,136.76	
Press <ENTER> to continue, M for Menu	
Enter amount of payments?	250
Total number of payments?	360
Number of payments per year?	12
Enter annual interest rate %?	18
Compounded how many times a year?	12
Value of annuity due: \$16,837.14	
Press <ENTER> to continue, M for Menu	

Line Number and Variable Cross-reference

Ref	Line Numbers
+-----+	
:00900:	965
:09000:	940
:10000:	935 965
:C :	930 935 9000 9005
:	Compounding periods per year
:CP :	9005 9015
:	Compounds per payment
:IC :	9000 9015
:	Interest rate/compound period
:IP :	950 9020
:	Interest per payment
:IR :	925 9000
:	Annual interest rate
:N2 :	920 935 9005
:	Number of payments per year
:N3 :	915 945
:	Number of payments
:P# :	950 955 960
:	Annuity Due value
:PQ :	910 950 955
:	Payment amount
:QV :	9015 9020
:	Effective interest rate
:X :	945 955 9015
:	Loop counter
+-----+	

Annuity Due— Present Value (I,PMT,N = PV) Program

```

900 CLS:PRINTTAB(5)"PRESENT VALUE OF AN
  ANNUITY DUE":REM A19
905 ' I.E. PAYMENTS MADE AT THE BEGINNING
  OF EACH MONTH
910 INPUT"Enter amount of payments";PQ
915 INPUT"Total number of payments";N3
920 INPUT"Number of payments per year";N
  2
925 INPUT"Enter annual interest rate %";
  IR
930 INPUT"Compounded how many times a year";C
935 IF N2>C THENPRINT"You may not make more
  than one deposit per interest compounding
  period.":GOSUB10000
940 GOSUB9000:' FIND EFFECTIVE INTEREST
  RATE PER PAYMENT
945 FOR X=N3-1TO 1 STEP -1
950 P#=(PQ+P#)/(1+(IP/100)):'MOVE BACK ONE
  PMT
955 NEXTX:P#=P#+PQ
960 PRINT"Value of the annuity due:";USING"$$,#####.##";P#
965 GOSUB10000:RUN900
9000 IC=(IR/C)/100:'INTEREST RATE PER COMPOUNDING
  PERIOD
9005 CP=C/N2:'COMPOUNDS PER PAYMENT
9010 'FIND EFFECTIVE INTEREST RATE FOR C2 PERIODS
9015 QV=1:FOR X=1 TO CP:QV=QV+(QV*IC):NEXT
  X
9020 IP=(QV-1)*100:'IP=INTEREST PER PAYMENT
9025 RETURN
10000 PRINT@281,"Press <ENTER> to continue, M for MENU";
10005 A$=INKEY$:IFA$="M"ORAS$="m"THENMAXFILES=0:
  CLEARO:MENUELSEIFA$<>CHR$(13)THEN
  10005ELSERETURN

```


ANNUITY DUE—PAYMENT ***(PV,N,I = PMT)***

If you are considering leasing a piece of equipment with a purchase price of \$1,000, and your intracompany cost of capital, or return on investment, is 18 percent per year, compounded monthly, what is the maximum you could afford to pay to lease the item?

The program takes the present value, total number of payments, number of payments per year, annual interest rate, and compounding period of the annuity due and calculates the payment.

Examples

<i>Program</i>	<i>Your response</i>
Enter present value of annuity?	1000
Total number of payments?	12
Number of payments per year?	12
Enter annual interest rate %?	18
Compounded how many times a year?	12
Working	
Payments would be \$90.33	
Press <ENTER> to continue, M for Menu	
Enter present value of annuity?	500
Total number of payments?	36
Number of payments per year?	12
Enter annual interest rate %?	6
Compounded how many times a year?	12
Working	
Payments would be \$15.14	
Press <ENTER> to continue, M for Menu	

Line Number and Variable Cross-reference

Ref	Line Numbers
+-----+	
:01000:	1100
:01055:	1080 1085
:01095:	1075
:01100:	1090
:09000:	1045
:10000:	1100
:C :	1030 9000 9005
:	Compounding periods per year
:CP :	9005 9020
:	Compounding periods per payment
:IC :	9000 9020
:	Interest rate/compounding period
:IP :	1060 9025
:	Interest per payment
:IR :	1025 9000
:	Annual interest rate
:N2 :	1020 9005
:	Number of payments per year
:N3 :	1015 1050 1055
:	Total number of payments
:P# :	1060 1070 1075 1080 1085
:	Miscellaneous variable
:PP :	1010 1050 1075 1080 1085
:	Estimated payment
:PQ :	1050 1060 1070 1080 1085
:	1095
:	Trial payment
:QQ :	1040 1080 1085
:	Trial increment
:QV :	9020 9025
:	Interest rate/compounding period
:X :	1055 1065 9020
:	Loop counter
+-----+	

Annuity Due—Payment (PV,N,I = PMT) Program

```
1000 CLS:PRINT"    PAYMENT FOR AN ANNUITY
      DUE":REM A20
1010 INPUT"Enter present value of annuit
y";PP
1015 INPUT"Total number of payments";N3
1020 INPUT"Number of payments per year";
N2
1025 INPUT"Enter annual interest rate %"
;IR
1030 INPUT"Compounded how many times a y
ear";C
1035 PRINTTAB(15)"Working."
1040 QQ=.1
1045 GOSUB9000:' FIND EFFECTIVE INTEREST
      RATE PER PAYMENT
1050 PQ=PP/N3
1055 FOR X=N3-1TO 1 STEP -1
1060 P#=(PQ+P#)/(1+(IP/100)):'MOVE BACK
ONE PMT
1065 NEXT X
1070 P#=P#+PQ
1075 IFINT(P#*100)=PP*100 THEN 1095
1080 IF P#<PP THEN PQ=PQ+(PQ*QQ):P#=0:GO
TO1055
1085 IF P#>PP THEN QQ=QQ*.1:PQ=PQ-(PQ*(Q
Q*10)):P#=0:GOTO 1055
1090 GOTO1100
1095 PRINT@240,"Payments would be:";USIN
G"$$#,#####.##";PQ
1100 GOSUB10000:RUN1000
9000 IC=(IR/C)/100:'INTEREST RATE PER CO
MPOUNDING PERIOD
9005 CP=C/N2:'COMPOUNDS PER PAYMENT
9010 'FIND EFFECTIVE INTEREST RATE FOR C
2 PERIODS
9015 QV=1:FOR X=1 TO CP:QV=QV+(QV*IC):NE
XT
9020 IP=(QV-1)*100:'IP=INTEREST PER PAYM
ENT
9025 RETURN
```



```
10000 PRINT@281,"Press <ENTER> to contin  
ue, M for MENU";  
10005 A$=INKEY$:IFA$="M"ORA$="m"THENMAXF  
ILES=0:CLEARO:MENUELSEIFA$<>CHR$(13)THEN  
10005ELSERETURN
```


ANNUITY DUE— INTEREST RATE, PV KNOWN (PV,PMT,N = I)

This program will determine the interest rate being charged for an annuity due, given the amount of the payments, their total number, the number per year, and the annuity's present value. The only assumption is that the compounding period and the payment period are the same; that is, if the payment is monthly, then the compounding period is also monthly. Use the program Equivalent Interest—Comparing Compound Periods (presented earlier in this chapter) to calculate different compounding periods.

Another use for this program is in comparing an installment plan with a simple bank loan or even paying cash for the item.

For example, if you are being asked to pay \$15.14 per month on a three-year equipment lease, where the equipment has a purchase price of \$500, what interest rate are you being charged?

Examples

<i>Program</i>	<i>Your response</i>
Enter amount of payments?	15.14
Total number of payments?	36
Number of payments per year?	12
Enter annuity present value?	500
Interest rate is .52% per period, 6.24% per year.	
Press <ENTER> to continue, M for Menu	
Enter amount of payments?	90.33
Total number of payments?	12
Number of payments per year?	12
Enter annuity present value?	1000
Interest rate is 1.52% per period, 18.24% per year.	
Press <ENTER> to continue, M for Menu	

Line Number and Variable Cross-reference

Ref	Line Numbers
+-----+	
:01150:	1225
:01190:	1200
:01205:	1190
:01225:	1215
:01230:	1185 1200
:10000:	1225
:IC :	1180 1195 1205 1210
:	Trial increment
:IP :	1180 1195 1205 1210 1215
:	1220 1235
:	Interest rate per period
:N2 :	1170 1220
:	Payments per year
:N3 :	1165 1230
:	Total number of payments
:P# :	1180 1190 1200 1235 1240
:	Miscellaneous variable
:PQ :	1160 1235 1240
:	Amount of payments
:PV :	1175 1190
:	Present value
:T :	1180 1205
:	Loop counter
:X :	1230 1240
:	Loop counter
+-----+	

Annuity Due— Interest Rate, PV Known (PV,PMT,N = I) Program

```

1150 CLS:PRINT"    INTEREST RATE FOR AN A
NNUITIY DUE":REM A21
1160 INPUT"Enter amount of payments";PQ
1165 INPUT"Total number of payments";N3
1170 INPUT"Number of payments per year";
N2
1175 INPUT"Enter annuity present value";
PV
1180 IP=0:IC=1:FORT=1 TO 3:P#=1
1185 GOSUB1230
1190 IFP#<PV THEN 1205
1195 IP=IP+IC
1200 P#=1:GOSUB 1230:GOTO 1190
1205 IP=IP-IC:IC=IC/10:NEXT T
1210 IC=IC*10:IP=IP+IC
1215 IF IP<0 THEN PRINT"DATA ERROR. PLEA
SE CHECK YOUR VALUES":GOTO1225
1220 PRINT"Interest rate is";IP;"% per p
eriod, ":PRINTUSING"###.##";IP*N2;:PRINT
"% per year."
1225 GOSUB10000:RUN1150
1230 FOR X=N3-1TO 1 STEP -1
1235 P#=(PQ+P#)/(1+(IP/100)):MOVE BACK
ONE PMT
1240 NEXTX:P#=P#+PQ
1245 RETURN
10000 PRINT@281,"Press <ENTER> to contin
ue, M for MENU";
10005 A$=INKEY$:IFA$="M"ORA$="m"THENMAXF
ILES=0:CLEARO:MENUELSEIFA$<>CHR$(13)THEN
10005ELSERETURN

```


ANNUITY DUE— INTEREST RATE, FV KNOWN (FV,PMT,N = I)

If you know the future value of an annuity due, as well as the payment amount, number of payments, and number per year, this program will tell you the interest rate.

Examples

<i>Program</i>	<i>Your response</i>
Enter amount of payments?	471.79
Total number of payments?	120
Number of payments per year?	12
Enter the future value of the annuity?	95000
Interest rate is .80% per period, 9.60% per year, compounded 12 times a year.	
Press <ENTER> to continue, M for Menu	
Enter amount of payments?	37.07
Total number of payments?	24
Number of payments per year?	12
Enter the future value of the annuity?	1000
Interest rate is .92% per period, 11.04% per year, compounded 12 times a year.	
Press <ENTER> to continue, M for Menu	

Line Number and Variable Cross-reference

Ref	Line Numbers
+-----+	
:02400:	2475
:02440:	2450
:02455:	2440
:02475:	2465
:02480:	2435 2450
:10000:	2475
:FV :	2425 2440
:	Future value
:IC :	2430 2445 2455 2460
:	IP calculation variable
:IP :	2430 2445 2455 2460 2465
:	2470 2485 2495
:	Interest rate/compounding period
:N2 :	2420 2470
:	Payments per year
:N3 :	2415 2480
:	Total number of payments
:P# :	2430 2440 2450 2485 2495
:	Trial principal value
:PQ :	2410 2485
:	Payment amount
:T :	2430 2455
:	Loop counter
:X :	2480 2490
:	Loop counter
+-----+	

Annuity Due— Interest Rate, FV Known (FV,PMT,N = I) Program

```

2400 CLS:PRINT"  INTEREST RATE OF AN AN
NUITY DUE":REM A22
2405 PRINTTAB(10)"FUTURE VALUE GIVEN
2410 INPUT"Enter amount of payments";PQ
2415 INPUT"Total number of payments";N3
2420 INPUT"Number of payments per year";
N2
2425 INPUT"Enter the future value of the
annuity";FV
2430 IP=0:IC=1:FORT=1 TO 3:P#=1
2435 GOSUB2480
2440 IFP#>FV THEN 2455
2445 IP=IP+IC
2450 P#=1:GOSUB 2480:GOTO 2440
2455 IP=IP-IC:IC=IC/10:NEXT T
2460 IC=IC*10:IP=IP+IC
2465 IF IP<0 THEN PRINT"DATA ERROR. PLEA
SE CHECK YOUR VALUES":GOTO2475
2470 CLS:PRINT"Interest rate is:":PRINTU
SING"###.##% per period, ###.##% per yea
r, compounded ### times a year.";IP;IP
*N2;N2
2475 GOSUB10000:RUN2400
2480 FOR X=1 TO N3
2485 P#=PQ+(P#+(P#*((IP)/100)))
2490 NEXT X
2495 P#=P#+(P#*((IP)/100)):RETURN
10000 PRINT@281,"Press <ENTER> to contin
ue, M for MENU";
10005 A$=INKEY$:IFA$="M"ORA$="m"THENMAXF
ILES=0:CLEARO:MENUELSEIFA$<>CHR$(13)THEN
10005ELSERETURN

```

5

Programming Utilities

The next two programs are the only ones that don't follow the numerical sequence order of the rest of the programs in this book: they both start with line number 10. This is because they are intended as programming aids, to be used independently.

The first program simply scans an ASCII-saved (.DO file) program and removes the unnecessary REM statements, freeing up valuable RAM for other programs and files.

You can use the second program to remove unneeded programs and files, without having to type KILL "filename" several times when you want to make more room available in your computer.

PACKER

The programs in this book, and many of the commercially available programs, have *REM statements* (remarks) in them. These statements are not used by the computer but are for the purpose of helping the programmer understand the inner workings of the program.

For example, in the program PACKER, there are a number of sentences that are preceded by the apostrophe ('). This symbol, which has the same effect as the word REM, tells the computer to ignore all that follows on that line. These sentences, however, will be valuable to you if you want to know how the program works or want to make changes in the program.

Space, though, is at a premium in the Model 100 computer. You may want to save space by deleting all unnecessary spaces, tabs, and REM statements. This can be done automatically by this program, PACKER.

To use PACKER, type it in as shown (you do not need to type in the apostrophes or the text that follows them). Save the program by typing SAVE"PACKER".

Now load in the BASIC program you wish to pack. Save that program in ASCII (.DO) format. If the program name is TEST1.BA, then save it by typing SAVE"TEST1.DO".

Now load and run PACKER. It will ask for the source program; type it in (for example, TEST1).

Then the program will ask for the name of the file you wish to create. Type in a different filename. The .DO at the end of the filename is optional. The program will add it if you omit it.

PACKER now starts reading your source program and printing the text on the new, shorter program that it is creating. When the new program is complete, PACKER will let you know how much memory it released from the source program.

PACKER will make the following changes to your programs:

1. All unnecessary spaces will be deleted. This line:

```
10 FOR X = 1 TO 10 : PRINT "HI THERE"
```

will be modified to read:

```
10 FORX=1TO10:?"HI THERE"
```

2. Tabs, if any, will be deleted.
-

3. Text preceded by an apostrophe will be deleted.
4. Text preceded by *REM* or *rem* will be deleted.
5. Any and all text enclosed in quotes will not be changed.
The program line:

10 PRINT "This is not a REM statement because it
is enclosed in quotes"

will not be changed in any way.

6. If a line contains *only* a REM statement, the entire contents of the line will be replaced by an apostrophe. For instance:

10 REM This is a remark

will be modified to read:

10 '

The line that had the REM statement on it will not be deleted, so that, if the line is referred to in a GOTO, GOSUB, IF . . . THEN, or ELSE statement, the program will not crash.

Line Number and Variable Cross-reference

Ref	Line Numbers				
+-----+					
:00010:	185				
:00030:	220				
:00070:	80	85			
:00085:	105	120	135	140	
:00090:	80	95	130		
:00110:	80	120			
:00125:	80				
:00140:	80				
:00145:	25				
:00190:	145				
:00215:	195	200			
:00220:	35	50			
:10000:	185				
+-----+					
:A\$: 215				
:	Keyboard input				
:FN\$: 30	45	220		
:	Filename				
:I\$(: 125	135			
:	Previous input character				
:I2\$: 60	75			
:	Characters to be deleted				
:IN\$: 70	75	85	90	95
:	100	110	115	120	125
:	130				
:	File input				
:LN	: 65	85	100	105	110
:	135	140			
:	Line length control				
:R\$: 220				
:	Used in filename formatting				
:X	: 75	80			
:	INSTR variable				
:Y	: 85	105	110	135	170
:	180				
:	Compressed length				
:Z	: 70	90	115	165	180
:	Original length				
+-----+					

PACKER Program

```

10 CLS:PRINTTAB(15)"PACKER":REM PACKER
15 PRINT"***** Current files in memory
*****":FILES
20 MAXFILES=2
25 ONERRORGOTO145
30 INPUT"Enter .DO file to pack";FN$
35 GOSUB220
40 OPEN FN$ FOR INPUT AS 1
45 INPUT"Enter .DO file to create";FN$
50 GOSUB 220
55 OPEN FN$ FOR OUTPUT AS2
60 I2$=CHR$(9)+" '" +CHR$(34)+"Rr"+CHR$(1
3):' <tab><space><apost><"><R><r><CR>
65 LN=0
70 IN$=INPUT$(1,1):Z=Z+1:'
   get chr from source
75 X=INSTR(1,I2$,IN$):'
   check for one of the characters
80 ON X GOTO 70,70,90,110,125,125,140:'
   if <tab>or<space>, pass (goto 90)
85 PRINT#2,IN$;:PRINTIN$;:LN=LN+1:Y=Y+1:
GOTO70:' Character is O.K. put in new fi
le
90 IN$=INPUT$(1,1):Z=Z+1:' Chr is <apost
>, get next chr til CR
95 IFIN$<>CHR$(13)THEN90:'
   keep ignoring until end of line
100 IF LN<7 THEN IN$="'" +IN$:'
   if possible line number only...
105 Y=Y+1:LN=0:GOTO85:'
   print the CR (goto 120)
110 PRINT#2,IN$;:PRINTIN$;:LN=LN+1:Y=Y+1
:' Character is a <">, print it
115 IN$=INPUT$(1,1):Z=Z+1:'
   get next character
120 IF IN$=CHR$(34)THEN85ELSE110:'
   keep reading & printing until <">
125 I$(1)=IN$:IN$=INPUT$(2,1)
130 IF IN$="EM"ORIN$="em"THEN90:'
   Chrs are REM, keep going til CR
135 PRINT#2,I$(1);:PRINTI$(1);:LN=LN+1:Y
=Y+2:GOTO85:'                                pri
nt the R, next 2, and the rest
140 LN=0:GOTO 85:'                                end of line

```



```
145 IF ERR<>54 THEN 190
150 CLOSE
155 CLS
160 PRINT
165 PRINT"Program Length was";Z
170 PRINT"Program is now      ";Y
175 PRINT
180 PRINT"Your savings      ";Z-Y
185 GOSUB10000:RUN10
190 CLOSE
195 IF ERR=52 THEN PRINTFN$;" Not found"
:GOTO215
200 IF ERR=53 THENPRINT"Source and desti
nation files must have different names"
:GOTO215
205 ON ERROR GOTO 0
210 RESUME
215 INPUT"Press Enter";A$:RUN
220 R$=RIGHT$(FN$,3):IF(R$<>".DO"AND R$<
>".do")THENIF LEFT$(R$,1)<>"."THENFN$=FN
$+ ".do"ELSEBEEP:PRINT"Invalid Filename :
":PRINTFN$:PRINT"Please Re-enter":GOTO 3
0
225 RETURN
10000 PRINT@281,"Press <ENTER> to contin
ue, M for MENU";
10005 A$=INKEY$:IFA$="M"ORA$="m"THENMAXF
ILES=0:CLEARO:MENUELSEIFA$<>CHR$(13)THEN
10005ELSERETURN
```

PURGER

This program utility makes it simpler to remove unwanted files from the Model 100 memory. When the program starts, a list of all the current files in memory is displayed. Below them is printed the prompt *Kill ?* Type the name of the file you want deleted, in uppercase letters. Give the full name, including the designation of file type (.BA, .DO, or .CO). If you begin to type in lowercase, a message will appear telling you to press the CAPS LOCK key.

Each time a file is deleted, the program relists all the current files so you can see which ones are left. If you delete a BASIC file, the prompt *Press F4 to continue.* appears. This is because normal program execution is halted whenever a BASIC program is removed from the directory, and you must rerun the PURGER program to continue. F4 does this for you.

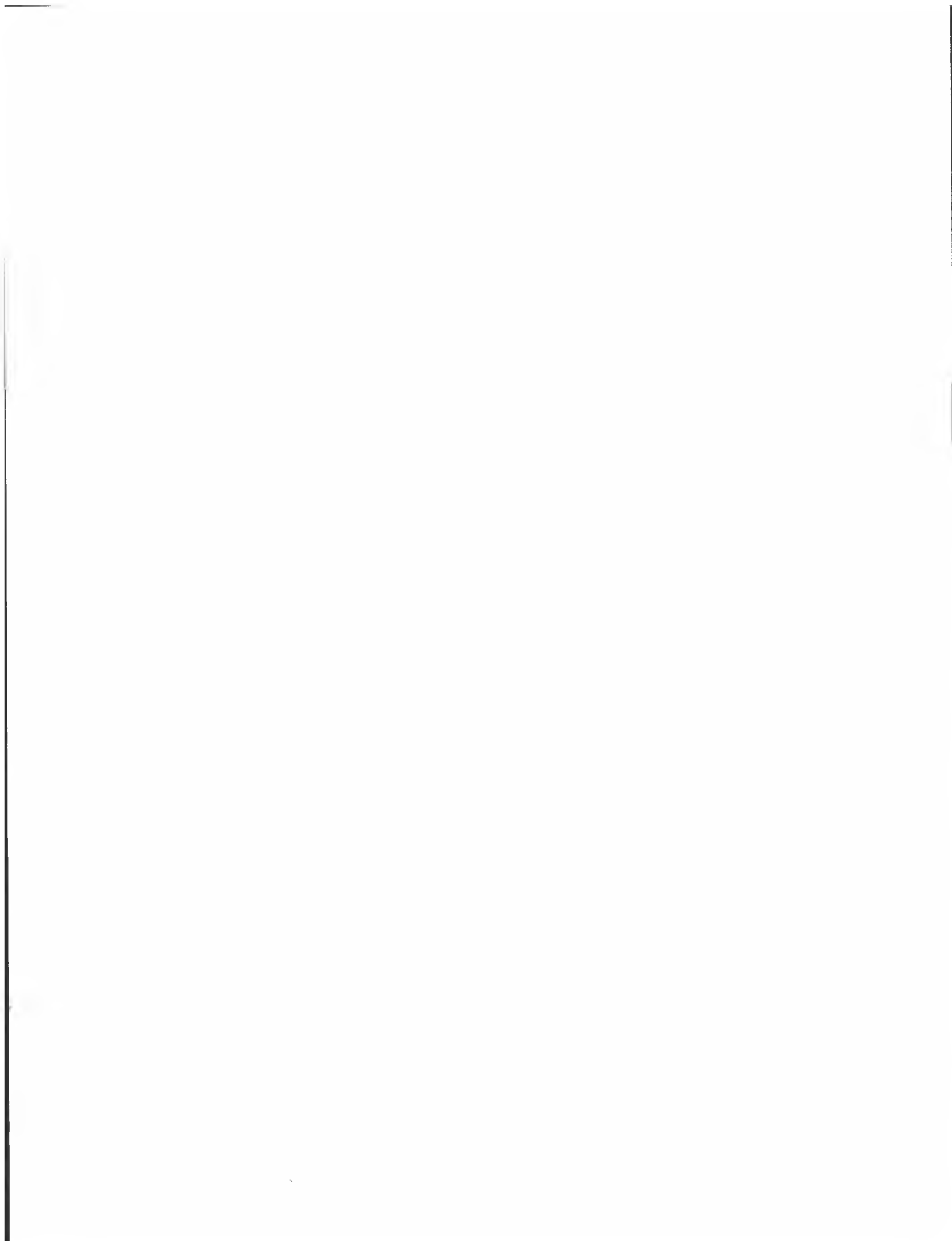
When you're finished, or if you have decided against deleting any programs, press the F8 key, and you'll be returned to the Model 100 menu.

Line Number and Variable Cross-reference

Ref	Line Numbers			
+-----+				
:00010:	90			
:00040:	60			
:00045:	100			
:00050:	50			
:00095:	15			
:00105:	95			
:I\$: 50	60	65	75
:	Letter input			
:X\$: 70	75	80	85
:	Filename			
:X1\$: 80			
:	File type			
+-----+				

PURGER Program

```
10 CLS: CLEAR 256: PRINT TAB(15) "PURGER": REM
   Purger
15 ON ERROR GOTO 95
20 KEY 4, "Run" + CHR$(13)
25 ON KEY GOSUB 0, 0, 0, 0, 0, 0, 0, 0, 110
30 KEY OFF: KEY(8) ON
35 FILES
40 PRINT @315, "Quit";
45 PRINT @240, "Kill ? ";
50 I$ = INKEY$: IF I$ = "" THEN 50
55 PRINT @280, "                                ";: PRINT
   T@247, " ";
60 IF I$ >= "a" AND I$ <= "z" THEN PRINT @280, "Hit
   <CAPS LOCK>";: GOTO 40
65 PRINT I$;
70 LINE INPUT X$
75 X$ = I$ + X$
80 X1$ = RIGHT$(X$, 3): IF X1$ = ".BA" THEN CLS: P
   RINT: PRINT: PRINT "Press <F4> to continue.
   "
85 KILL X$
90 RUN 10
95 IF ERR <> 52 AND ERR <> 55 THEN 105
100 BEEP: PRINT @280, "File not found";: PRI
   NT @247, "                                ";: RESUME 45
105 ON ERROR GOTO 0: RESUME
110 MAXFILES = 0: CLEAR 0: MENU
```

Select Bibliography

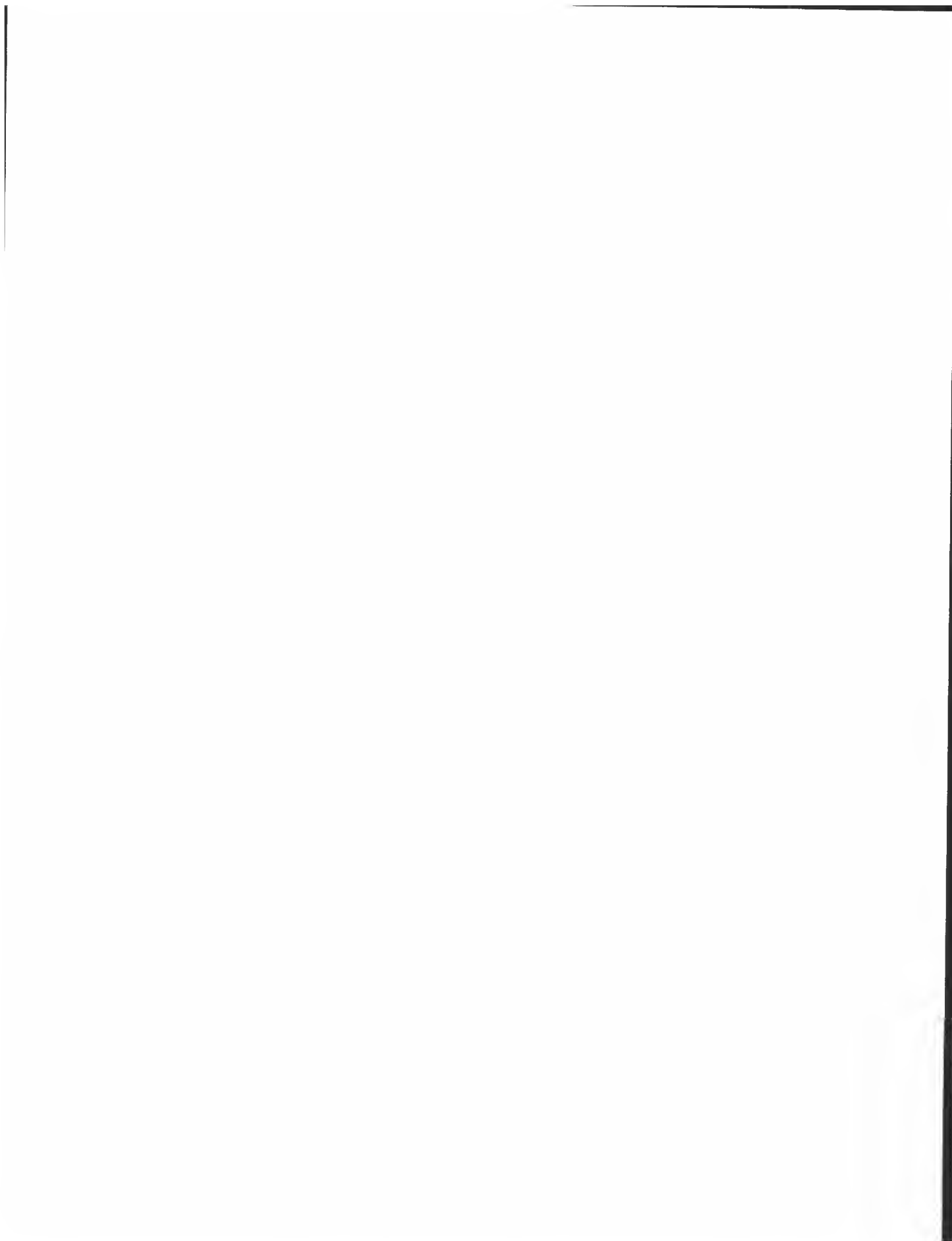
References we used in writing this book include:

Boone, Louis E., and David L. Kurtz. *Contemporary Marketing*. 3rd ed. Hinsdale, Ill.: Dryden Press, 1980.

Pyle, William W., John Arch White, and Kermit D. Larson. *Fundamental Accounting Principals*. 8th ed. Homewood, Ill.: Richard D. Irwin, Inc., 1978.

Texas Instruments Learning Center. *Calculator Analysis for Business and Finance*. Texas Instruments Inc., 1977.

Weston, J. Fred, and Eugene F. Brigham. *Essentials of Managerial Finance*. 5th ed. Hinsdale, Ill.: Dryden Press, 1979.



Glossary

Add-on interest. A form of simple interest as it applies to loan repayment. The stated interest rate is multiplied by the total amount of the loan each year.

Amortize. To liquidate on an installment basis. An amortized loan is one in which the principal amount of the loan is repaid in installments during the life of the loan.

Annuity. A series of payments of a fixed amount for a specified number of time periods.

Annuity due. An annuity for which payments are made at the beginning of each period. A common example of such a payment is a lease payment.
See Ordinary annuity.

Apostrophe ('). A BASIC statement meaning *remark*. All information that follows the apostrophe on that program line is ignored by the computer.
See REM.

Average collection period ratio. Receivables divided by sales per day. This ratio can evaluate your collections department and sales terms.

Balloon payment. The final payment, larger than preceding payments, made on a loan or debt that is not fully amortized.

Book value. The accounting value of an asset.

Calculation variable. A program variable that is used to hold the products of calculations. It does not necessarily represent a specific item such as *useful life* or *present value*.

CLEAR. A Model 100 BASIC command that erases unnecessary information from previous programs.

Compound interest. An interest rate that is applied to the sum of original principal plus interest previously earned.

Continuous compounding. The applying of interest at every point in time rather than at certain points in time (monthly or yearly, for instance). If continuous compounding applies, your money is growing every minute, every second, of every day.

Current ratio. Current assets divided by current liabilities. This is a common measure of short-term solvency.

Debt ratio. Total debt divided by total assets.

Debt-to-total-assets ratio. Total debt divided by total assets. This measures how much of the firm's value is provided by creditors.

Economic order quantity (EOQ). The optimum size of a merchandise order, calculated so that inventory is neither held too long nor ever out of stock.

Effective interest rate. The interest rate used in computation or evaluation. *See* Nominal interest rate.

Fixed asset turnover ratio. Sales divided by fixed assets. This ratio can indicate whether the firm is using its plant and equipment to their maximum capacity.

Fixed charge coverage ratio. Income available for meeting fixed charges, divided by fixed charges. This ratio can help determine the overall stability of the company and takes the widespread use of leasing into account.

Fixed charges. Costs that do not vary with level of output. For example, the lease costs of a manufacturing plant are the same regardless of whether

the plant is producing at full capacity or half capacity. *See also* Variable costs.

Future value. The amount that a payment will be worth in the future after interest is calculated. *See* Present value.

Income available for meeting fixed charges. The sum of profit before taxes, interest charges, and lease obligations.

Installment plan. Often called *revolving charge plan*, this is the type of loan repayment program that charges interest on the unpaid balance. Most credit cards use this type of interest.

Internal rate of return. The rate of return on an asset investment. The internal rate of return is calculated by finding the discount rate that equates the present value of future cash flows to the cost of the investment.

Inventory turnover ratio. Sales divided by inventory. This ratio can indicate whether a firm is overstocking or understocking merchandise.

Lifespan. Also called *useful life*, or simply *life*, of an asset, this is the length of time for which the asset is assumed to be productive. If a machine is expected to last five years, its lifespan is said to be five years. Note that this is an accounting value, not necessarily a realistic value. The life of all items is estimated for the purpose of depreciation.

Loop counter. A program variable used to hold the number of times that the program flow has passed a certain point in the program.

MAXFILES. A Model 100 BASIC command that specifies the number of input and output files you will be using. The command also erases unneeded information from previous programs.

Menu program. A program, or part of a program, that coordinates different or related programs or subprograms. A menu program allows the user to choose which program or function to use.

MERGE. A Model 100 BASIC function that allows you to combine an ASCII-saved file (one with a filename ending in .DO) with a BASIC program while it's in BASIC.

Miscellaneous variable. A program variable that is used to hold the products of calculations. It does not necessarily represent a specific item such as *useful life* or *present value*.

Net worth. The capital and surplus of a firm; the common shareholder's position.

Nominal interest rate. The contracted or stated interest rate. *See* Effective interest rate.

Ordinary annuity. An annuity for which payments are made at the end of each period. Common examples of such payments are loan payments and revolving charges. *See* Annuity due.

Periodic compounding rate. An interest rate that is continuously compounded.

Present value. The value today of a payment to be made in the future after interest is calculated. *See* Future value.

Prime rate. The lowest rate of interest that commercial banks will charge to supposedly risk-free customers.

Profit margin. The ratio of a firm's after-tax profits to its sales.

Profit margin on sales ratio. Net profit after taxes, divided by sales. This measures the profit, per dollar, of the sales.

Program gap. The shortage of programs to fill the needs of businesspeople and the capabilities of their computers.

Quick ratio. Also called the *acid test*. The difference between current assets and inventory, divided by current liabilities. This measures the ability of a firm to pay off its debt without the help of its inventory. This might be the case in the event of bankruptcy.

REM. A BASIC statement meaning *remark*. All information that follows the word REM on the program line is ignored by the computer. *See* Apostrophe (').

Required rate of return. The rate of return that stockholders expect to receive on common stock investments.

Return on net worth. Net profit after taxes, divided by net worth. This measures the rate of return on the stockholder's investment.

Return on total assets ratio. Net profit after taxes, divided by total assets. This measures the return on the total investment, or ROI, of the firm.

Salvage value. The value of a capital asset at the end of a specified period. This is the cost at which an asset will be sold when the firm is finished with it. It is used in depreciation calculations.

Standard deviation. A statistical measurement of the variability of a set of observations from the mean value of the distribution.

Subroutine. A smaller program that is used by a larger program. If, for example, your main program often needs to calculate the number of days between two dates, a separate program calculating days between dates is written so that it can be called several times by the main program.

Times interest earned ratio. The sum of profit before taxes and interest charges, divided by interest charges. This ratio measures the ability of a firm to withstand a sales slump.

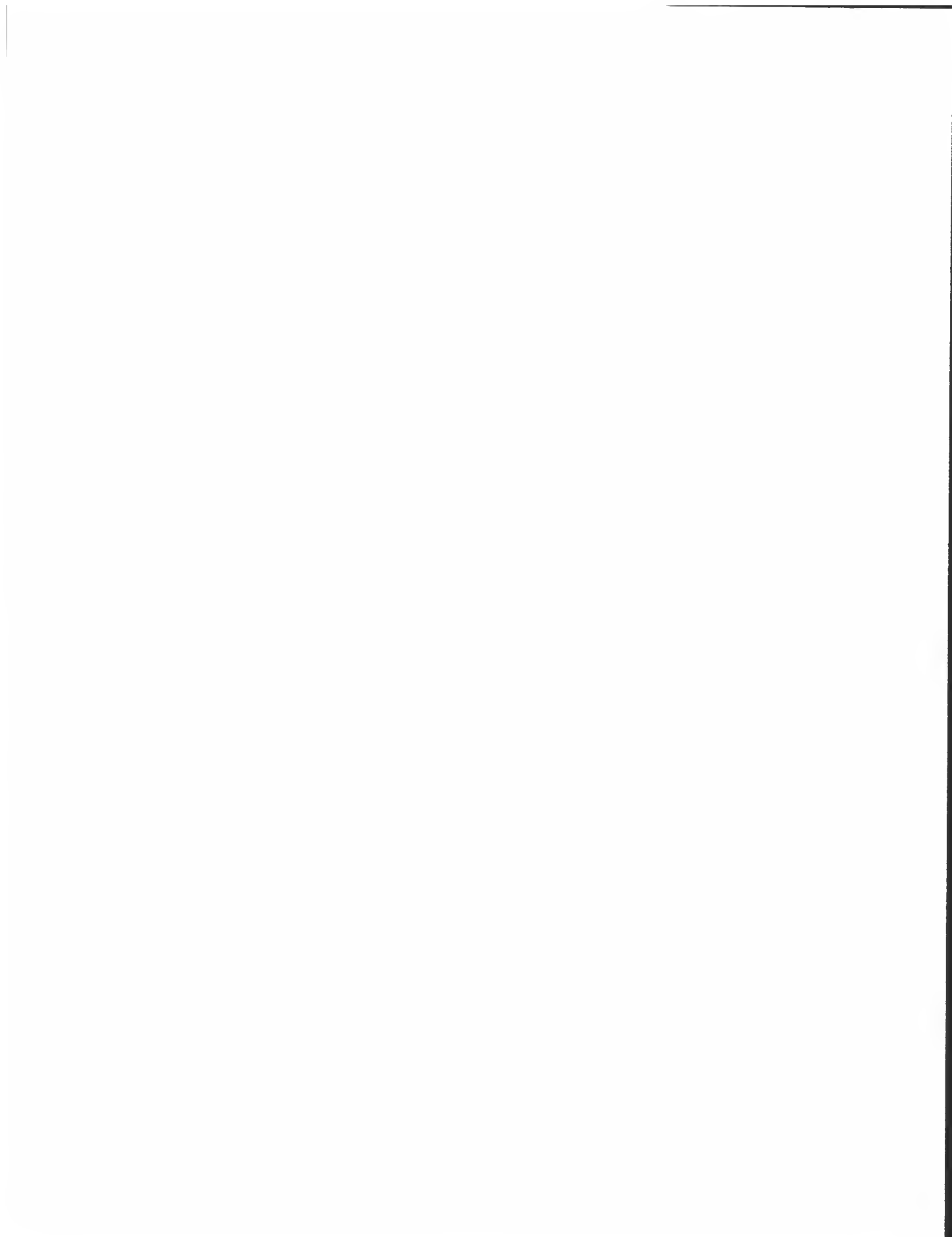
Total asset turnover ratio. Sales divided by total assets. This relates the volume of business to the asset investment.

Variable costs. Costs that increase according to increases in production. These include materials and maintenance. *See also* Fixed charges.

Weighted average. An average that is based on some factor in the items being averaged. Using a weighted average system, some items being averaged have more importance, or weight, than others.

Working capital. A firm's investment in short-term assets such as cash, short-term securities, accounts receivable, and inventories. Gross working capital is the total current assets. Net working capital is the current assets minus current liabilities.

Yield. The rate of return on an investment. *See* Internal rate of return.



Index

- A\$, xii
- Accountant's helpers, 141
- Adding, xi
- Add-on interest, 267
- Add-on interest, converting, to
annual interest, 79
- Algebraic calculators, 20
- Amortization, 158, 267
- Analysis, breakeven unit, 30
- Annual interest, converting to,
from add-on interest, 79
- Annuity due, 201, 267
- Annuity due, finding future value
of, 237
- Annuity due, finding interest rate
of, from future value, 252
- Annuity due, finding interest rate
of, from present value, 249
- Annuity due, finding payment of,
from present value, 245
- Annuity due, finding present
value of, 242
- Annuity, ordinary, 201, 267, 270
- Annuity, ordinary, finding future
value of, 202
- Annuity, ordinary, finding future
value of, with continuous
compounding of interest, from
payment, 208
- Annuity, ordinary, finding future
value of, with continuous
compounding of interest, from
present value, 205
- Annuity, ordinary, finding
interest rate of, from future
value, 227
- Annuity, ordinary, finding
interest rate of, from present
value, 230
- Annuity, ordinary, finding
number of payments of, 233
- Annuity, ordinary, finding
payment of, from future value,
224
- Annuity, ordinary, finding
payment of, from present
value, 220
- Annuity, ordinary, finding
present value of, 211
- Annuity, ordinary, finding
present value of, with
continuous compounding of
interest, from future value, 217
- Annuity, ordinary, finding
present value of, with
continuous compounding of
interest, from payment, 214
- Apostrophe, x, 267, 270

- Asset turnover ratio, fixed, 2, 268
Asset turnover ratio, total, 2, 271
Average collection period ratio, 2, 267
Average, weighted, 168, 271
Average yield of an investment, 164

Balance unpaid, 92
Balloon payment, 267
Bar chart, 133
Book value, 268
Borrowing money, 76
Breakeven unit analysis, 30

Calculation variable, 268
Calculator program, 20
Calendar, 58
Capital asset pricing model, 96
Capital, working, 271
Cash management model, 100
Charge coverage ratio, fixed, 2, 268
Charges, fixed, 3, 268
Chart, bar, 133
Chart, pie, 125
CLEAR, ix, 268
Collection period ratio, average, 2, 267
Comparing compound periods, 197
Compounding, continuous, 268
Compounding rate, periodic, 270
Compound interest, 268
Compound interest computations, 185
Compound interest, finding future value of, 181
Compound interest, finding interest rate of, 194
Compound interest, finding present value of, 190
Compound interest, finding time period of, 186
Compressing, xi, xii
Continuous compounding, 268
Converting add-on interest to annual interest, 79
Cost before markup, 110
Costs, fixed. *See* Charges, fixed
Costs, variable, 2, 269, 271

Counter, loop, 269
Credit, easy, 71
Credit plan, 71
Cross-reference, xi
Current ratio, 2, 268

Days between dates, 62, 168
Debt-to-total-assets ratio, 2, 268
Decision making, management, 65
Declining-balance depreciation, 150
Declining-balance versus straight-line depreciation, 155
Deleting, xi
Depreciation, 142
Depreciation, declining-balance, 150
Depreciation, straight-line, 143
Depreciation, straight-line versus declining-balance, 155
Depreciation, sum-of-year's-digits, 146
Deviation, standard, 271
Direct statement in file, x
Discounted price, 116
Discount, price before, 119
Dividends, 164

Earnings per share, 168
Easy credit, 71
Economic order quantity, 13, 268
Effective interest rate, 268, 270
EOQ. *See* Economic order quantity
Equivalent interest: comparing compound periods, 197

Fixed asset turnover ratio, 2, 268
Fixed charge coverage ratio, 2, 268
Fixed charges, 2, 269, 271
Fixed costs. *See* Fixed charges
Future value, 269, 270

Glossary, 267

Income available for meeting fixed charges, 269
Installment plan, 92, 269
Installment plan schedule, 92

- Insurance, term versus whole life, 66
Interest, add-on, 267
Interest: comparing compound periods, 197
Interest, compound, 268
Interest, compound, finding future value of, 181
Interest, compound, finding interest of, 194
Interest, compound, finding present value of, 190
Interest, compound, finding time period of, 186
Interest computations, compound, 185
Interest rate, effective, 268, 270
Interest rate, nominal, 268, 270
Interest, simple, finding future value of, 177
Interest, total, paid on a loan, 76
Internal rate of return, 269, 271
Inventory turnover ratio, 2, 269
Investment, average yield of, 164
Invoice program, 36

Job quote, 49

Lease versus purchase, 86
Life. *See* Lifespan
Lifespan, 142, 269
Linear calculators, 20
Loan repayment and rebate, 82
Loan, total interest paid on, 76
Loop counter, 269

Management decision making, 65
Management model, cash, 100
Margin, profit, 122, 270
Markup, 107
Markup, cost before, 110
Markup, percentage, 113
MAXFILES, ix, 269
Maximum line length, xii
Menu program, viii, 269
MERGE, viii, 269
Miscellaneous variable, 270
Model, cash management, 100
Money, borrowing, 76
Net worth, 2, 270
Net worth, return on, 2, 270
Nominal interest rate, 268, 270

Order quantity, economic, 130
Ordinary annuity, 201, 267, 270
Ordinary annuity, finding future value of, 202
Ordinary annuity, finding future value of, with continuous compounding of interest, from payment, 208
Ordinary annuity, finding future value of, with continuous compounding of interest, from present value, 205
Ordinary annuity, finding interest rate of, from future value, 227
Ordinary annuity, finding interest rate of, from present value, 230
Ordinary annuity, finding number of payments of, 233
Ordinary annuity, finding payment of, from future value, 224
Ordinary annuity, finding payment of, from present value, 220
Ordinary annuity, finding present value of, 211
Ordinary annuity, finding present value of, with continuous compounding of interest, from future value, 217
Ordinary annuity, finding present value of, with continuous compounding of interest, from payment, 214

PACKER, 256
Payment, balloon, 267
Percentage markup, 113
Period, average collection, ratio, 2, 267
Periodic compounding rate, 270
Pie chart, 125
Plan, installment, 269
Present value, 269, 270

- Price before discount, 119
 - Price before sales tax, 104
 - Price, discounted, 116
 - Pricing model, capital asset, 96
 - Prime rate, 270
 - Profit margin, 122, 270
 - Profit margin on sales ratio, 2, 270
 - Program gap, iii, 270
 - Purchase versus lease, 86
 - Purchase versus time payment, 71
 - PURGER, 261

 - Quick ratio, 2, 270
 - Quote, job, 49

 - Rate, internal return, 271
 - Rate of return, 164
 - Rate of return, required, 270
 - Rate, prime, 270
 - Ratio, current, 2, 268
 - Ratio, debt-to-total-assets, 2, 268
 - Ratio, fixed asset turnover, 2, 268
 - Ratio, fixed charge coverage, 2, 268
 - Ratio, inventory turnover, 2, 269
 - Ratio, profit margin on sales, 2, 270
 - Ratio, quick, 2, 270
 - Ratio, return on total assets, 2, 271
 - Ratios, 2
 - Ratio, times interest earned, 2, 271
 - Ratio, total asset turnover, 2, 271
 - Rebate and loan repayment, 82
 - REM, x, 267, 270
 - Required rate of return, 270
 - Return on net worth, 2, 270
 - Return on total assets ratio, 2, 271
 - Return rate, internal, 271

 - Sales ratio, profit margin on, 2, 270
 - Sales tax, price before, 104

 - Salvage value, 271
 - Shares outstanding, 168
 - Simple interest, finding future value of, 177
 - Standard deviation, 271
 - Stock share earnings, weighted average, 168
 - Straight-line depreciation, 143
 - Straight-line versus declining-balance depreciation, 155
 - Subroutine, 62, 168, 271
 - Sum-of-year's-digits depreciation, 146

 - Term versus whole life insurance, 66
 - Time payment versus purchase, 71
 - Times interest earned ratio, 2, 271
 - Total asset turnover ratio, 2, 271

 - UL error, xi
 - Unpaid balance, 92
 - Useful life, 269
 - Using this book, viii

 - Value, book, 268
 - Value, future, 269, 270
 - Value, present, 269, 270
 - Value, salvage, 271
 - Variable, calculation, 268
 - Variable costs, 2, 269, 270
 - Variable, miscellaneous, 270
 - Variables, xi

 - WAIT, ix
 - Weighted average, 168, 271
 - Weighted average number of stock shares outstanding, 168
 - Whole versus term life insurance, 66
 - Working capital, 271

 - Yield, 271
 - Yield rate, 164
-

60 BUSINESS APPLICATIONS PROGRAMS

> \$17.95

FOR THE TRS-80 MODEL 100 COMPUTER

When you own a computer as powerful as the TRS-80 Model 100—*InfoWorld's* Hardware Product of the Year for 1983—you need programs that will make the most of it. This book gives you 60 programs you can put to work immediately in your business and in your personal financial planning.

Designed by a businessperson and a professional writer-programmer, these programs quickly perform a wide variety of useful business calculations. Use this book to

- calculate 12 business analysis ratios
- calculate breakeven sales levels
- create and print invoices
- calculate depreciation and amortization
- find the average yield of your investments
- create bar and pie charts

and much more!

This book couldn't be easier to use—just type in the programs you need. All programs are fully explained, and you'll see each program listing exactly as it will appear on your computer screen.

Turn your Model 100 into a valuable business tool with **60 BUSINESS APPLICATIONS PROGRAMS FOR THE TRS-80 MODEL 100 COMPUTER.**

Terry Kepner is a professional writer and programmer. He currently writes monthly columns for *80 Microcomputing*, and *Portable 100 Magazine*, and reviews hardware and software for *Byte*, *Popular Computing*, *Microcomputing*, *Hot Coco*, and others.

Mark Robinson works for RAMparts Publishing, a company that owns and markets numerous computer mailing lists. He previously was president of Interpro Corporation, which published software for the TRS-80 Models I and III, and served as a computer consultant to businesses.

Scott, Foresman and Company

ISBN 0-673-18052-2